Modelling Visual Objects Regardless of Depictive Style

Wu, Qi

Award date:
2015

Awarding institution:
University of Bath

Modelling Visual Objects Regardless of Depictive Style

submitted by

Qi Wu

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Sciences

December 2014

ABSTRACT

Visual object classification and detection are major problems in contemporary computer vision. State-of-art algorithms allow thousands of visual objects to be learned and recognized, under a wide range of variations including lighting changes, occlusion and point of view etc. However, only a small fraction of the literature addresses the problem of variation in depictive styles (photographs, drawings, paintings etc.). This is a challenging gap but the ability to process images of all depictive styles and not just photographs has potential value across many applications. This thesis aims to narrow this gap.

Our studies begin with *primitive shapes*. We provide experimental evidence that primitives shapes such as 'triangle', 'square', or 'circle' can be found and used to fit regions in segmentations. These shapes corresponds to those used by artists as they draw. We then assume that an object class can be characterised by the qualitative shape of object parts and their structural arrangement. Hence, a novel *hierarchical graph representation* labeled with primitive shapes is proposed. The model is learnable and is able to classify over a broad range of depictive styles. However, as more depictive styles join, how to capture the wide variation in visual appearance exhibited by visual objects across them is still an open question. We believe that the use of a graph with *multi-labels* to represent visual words that exists in possibly discontinuous regions of a feature space can be helpful.

ACKNOWLEDGEMENTS

Three years ago, when my supervisor Dr. Peter Hall asked me: 'why do you want to be a PhD ?' in my PhD candidate interview, I said:' I am so interested in Computer Vision and I want to be an expert in this area. To become a PhD means I have the opportunity to study the most cutting-edge knowledge in this subject.' This might not be the best answer and I might still can not give the best answer even now. However, I am happy to say I still maintain the enthusiasm to the Computer Vision and I want to continue my research. I don't believe this thesis is the end. It is just a new start.

At first, I would like to thank my supervisor, Dr. Peter Hall, not only for providing research advices, but also for supporting my life. In the study and research process, Peter always could give me useful suggestions. He cared about my research progress very much. We had our regular meetings each week in the past three years and I always can find the opportunity to talk with him, to discuss the latest progress and receive the feedback. Moreover, his rigorous academic attitude influenced me a lot. He always reminded me to think over the problem as a scientist. He also cared about me in life. As an overseas student, the life was not easy. Peter always were ready to help and considered my feelings. I am so lucky he can be my supervisor and received his support.

I also would like to thank my colleague, the research associate, Dr. Hongping Cai, for many useful suggestions in my research. We were working on the same project and she gave me many helpful ideas. Discussions with her not only helped me a lot in this thesis, but also gave me many good advices on my academic career. I wish everything goes well for her and her baby.

Moreover, thanks to my colleagues and friends, Dr. Chuan Li, Mr. Kewei Duan, Mr. Rui Tang and Mr. Yifei Wang. They gave me a lot of interesting ideas about this thesis. At the same time, thank them to give me so much help during my academic and daily life.

Additionally, I also want to thank my wife, Mrs. Yixin Wang. In the past three years, she used her love and understanding to fully support me and our family. To help me to focus on my study, she undertook nearly all the housework. And she brings so many supervise and happiness to my life. Without her support, I cannot finish my

PhD so quickly. I also want to give my deepest gratitude to my parents. They gave me huge support in my study and life. Without their economic and spiritual help, I cannot finish my study.

Finally, I want to thank my examiners, Dr. Matthew Brown and Prof. Roberto Cipolla, who gave me so many helpful suggestions about this thesis and further researches.

Thank you. Thank everyone who helps me in my study and life.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

Object recognition is one of the most fascinating abilities that humans possess. With a simple glance at an object, humans are able to tell its identity or category despite of the appearance variation due to change in pose, illumination, texture, deformation, and under occlusion. Researchers are trying to push computer vision algorithms to achieve human performance. However, a significant area is overlooked – humans are able to recognise, locate and classify objects in a seemingly unlimited variety of depictions: in photographs, in line drawings, as cuddly toys, in clouds.

In this thesis, we show it is possible to learn models of object classes that generalise across different depictive styles. More specifically, we test the hypothesis that:

> ***object class representation is the key to solve the cross-depiction object recognition problem.***

Visual object class modelling has been studied for many years. Significant effort has been paid to developing representational schemes and algorithms aimed at recognising objects in photographs. However, nearly all contemporary methods are premised upon low variance in object features, which explains a significant drop in performance for cross-depiction problems where the variance in features is wide.

To solve this problem we posit the use of shape, structure and multi-labels as representational elements with which to describe a visual object class. Shape is a natural representational element in art since artists draw initial sketches using simple shapes, to lay out objects and scenes. We argue that simple primitive shapes are common properties existing in both photo realistic images and paintings, thus they can be used as a robust representation for cross-depiction modelling. Based on this assumption, we test the hypothesis that a significant fraction of regions in segmentations can be fitted by *primitive shapes*, such as 'triangle', 'square', or 'circle'. Structural is also an important global information which can not be ignored and it has been used in many

(a) Lascaus (b) Bison (c) Petroglyphs

Figure 1-1: *Some examples of cave painting. (a).Image of a horse from the Lascaux caves, stone age. (b)Reproduction of a bison of the cave of Altamira. (c). Petroglyphs, from Sweden, Nordic Bronze Age.*

modelling methods. We then assume that an object class can be characterised by the qualitative shape of object parts and their structural arrangement. Hence, a novel *hierarchical graph representation* labeled with primitive shapes is proposed. However, as more depictive styles are included, the local visual appearance variation becomes wider. This variation is typically much wider than for lighting and viewpoint variations usually considered for photographic images. How to capture the wide variation in visual appearance exhibited by visual objects across them is still an open question. We argue that the use of a graph with *multi-labels* to represent visual words that exists in possibly discontinuous regions of a feature space is of value and is more effective than attempting characterize all the depictive styles in a monolithic model.

1.1 Motivation

The motivation of this thesis is not only to fulfil the gap of literatures (there is little researches about cross-depiction problem, see Chapter 2.), but also for both scientific and practical reasons.

Object recognition is a topic that has received continuous and consistent attention within computer vision, pattern recognition and machine learning. It has been widely used in many fields. For example, in the surveillance area, Content Based Image Retrieval (CBIR) area etc. The neurobiologist Marr [95] claims that vision problems can be attributed to 'What is Where' - what object is in which place. This is the motivation of object recognition and it is concerned with determining the identity of an object being observed in the image.

Painting has already appeared since the beginning of human civilization. Approximately 40000 years ago, prehistoric people draw on rock, known as 'Cave Painting'. People, animals and natural elements are the main contents in their paintings. Figure 1-1 shows some examples of 'Cave Painting'. Humans still are able to recognise objects in these paintings even after dozens of centuries. Since then, with the change over age

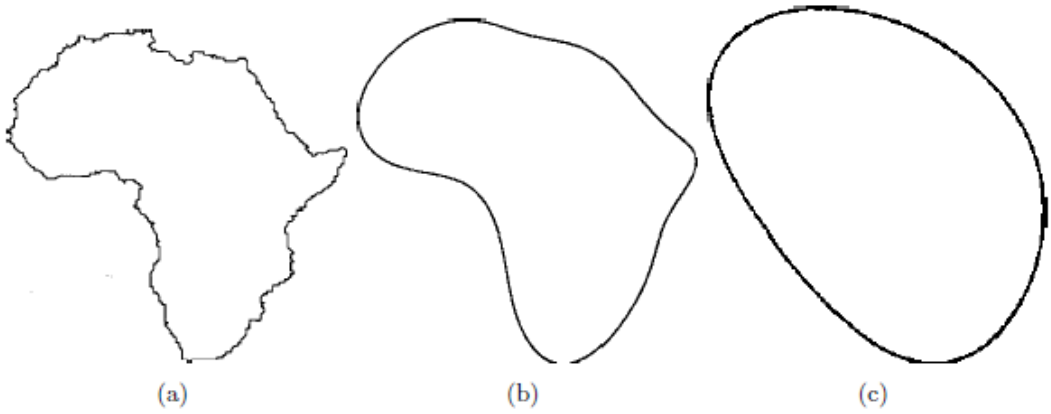


Figure 1-2: *An example of simple visual abstraction performed by reducing the overall curvature of the contour of Africa.*

and region, styles of painting various a lot. According to [158], painting styles can be classified based on different rules, such as areas, years and schools. In the East, we have Chinese painting [152], Japanese Painting[154] Korea Painting[155] and Indian painting etc. In the West, there are Modernism[156], Impressionism[153], Abstract styles[151], Outsider art[157], Photorealism[159], Surrealism[160] and so on. The number of styles are still increasing but interestingly, people always can recognise the objects in most of these styles. Before the birth of camera (the first camera was made in 1839), painting is the only way to record objects and events and people are still drawing nowadays. However, as a such important source in human history and daily life, painting is ignored by the computer vision, especially for object recognition. There are many reasons to introduce paintings into object recognition.

1.1.1 Scientific Motivation

The scientific motivation for introducing paintings or art in computer vision is that art is a visual abstraction that is parsimonious yet meaningful: that is, art works can and do use very little information to represent things in such a way they can be recognised. Parsimonious descriptions are a great value because they are efficient to store and use, they tend to be robust and generalise well. In other words, they are the natural models of objects.

Artists are experts in translating their observation, imagination and knowledge in an abstract manner. They are able to convey visual expression through forms of paintings, and are extremely good in delivering abstraction in their work. Famous artists such as Van Gogh, Picasso to name a few, have created paintings that are not only aesthetic, but also tell a story at a high level of abstraction. Many artists begin their paintings with shapes or building blocks that eventually turn into masterpieces of fine detail.

The importance of cave art is that it shows the symbolisms used (ie the abstractions)

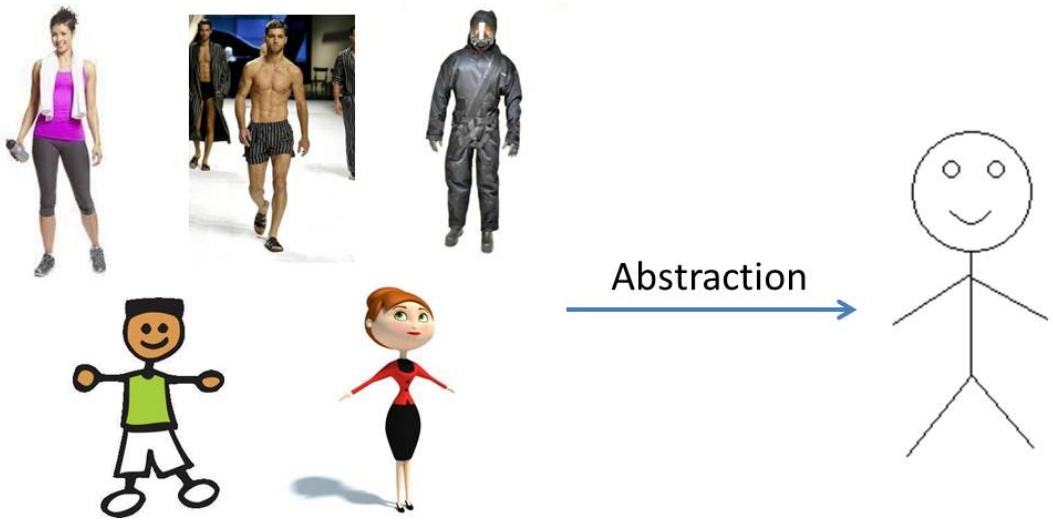


Figure 1-3: *Various styles can be abstracted as a stick man.*

have not changed over tens of thousands of years. This is obvious when one considers a child's drawing of a car in which all four wheels are shown – the child draws what they know of a car, not what is seen. In addition, a line drawing, for example, is much more compact in terms of information content than a photograph – drawings are abstractions in the sense that a lot of data is discarded, but information germane to the task of recognition is (typically) kept. This suggests that visual class models used in computer vision should exhibit a similarly high degree of abstraction.

Abstraction can be formed by reducing the information content of a concept or an observable phenomenon, typically to retain only information which is relevant for a particular purpose. The visual abstraction then can be defined as the process of reducing the visual content of a given object progressively. For example, the contour of the African continent can be abstracted by iteratively reducing the curvature of the contour, as shown in figure 1-2. It can be seen that the contour reduces from a complex structure with lots of jagged edges, to an ellipse-like shape. Although there is significant loss of information, the shape in figure 1-2 (c) can be refereed to as a crude approximation of that in figure 1-2 (a).

The importance of visual abstraction was also underlined by Picasso, when he described art as "the elimination of the unnecessary". Even though artwork is sometimes highly abstracted, humans can still correctly perceive the objects in them and often interpret the intended underlying meaning. It could be that art is a visual representation of the way our brains encode the visual world. Moreover, the encoding is possible a generative model. Hence, introducing paintings into current computer vision researches has the potential value in a scientific view. It forces researchers to design the recognition system via studying how people understand the objects and motions depicted in

(a) Centaurs (b) Pegasus (c) Medusa

(d) Minotaur (e) Griffin (f) Sphinx (g) Siren

Figure 1-4: *Some examples of mythological creatures. These objects only can be observed in art works.*

art. In other words, it is the art to let people know that the abstraction can lead to the robustness to non-salient variation and makes us to consider using simple primitive shapes and structures to represent object class regardless of depictive styles. Figure 1-3 shows an example that various style people images can be abstracted as a stick man.

A second reason for being interested in extending the gamut of depictions available to computer vision is that not all visual objects exist in the real world. Mythological creatures (Figure 1-4), for example, have never existed but are recognisable nonetheless. Most of these objects only exist in paintings and artworks. If computer vision is to recognise such visual objects it must emulate the human capacity to disregard depictive style with respect to recognition problems. Some similar situations also can be observed in photos, for example, a man with a cartoon mask. How to detect and recognise these assembly objects is still an open question and which is related to model objects class across different depictive styles, since a more generative model might be required.

1.1.2 Practical Motivation

There are also many practical reasons for wanting visual class objects that generalise across depictions. One reason is that computer vision should not discriminate between visual class objects on the basis of their depiction - a face is a face whether photographed or drawn. Given a picture of someone's face as a query, a search over a database of images should ideally return all portraits of them, no matter what style. Figure 1-5

Figure 1-5: *Example faces depicted in various styles.*

shows more faces depicted in various styles while figure 1-6 shows some failing cases when using 'Google Image' to retrieval face images that depicted in different styles. It is clearly shown that top responses are not faces, although they may have similar colour pattern with the query images.

Non-photorealistic rendering (NPR) also can be benefit from introducing art into object modelling, for example, synthesis art work from photographs using the object class model, especially for the abstraction art. Song et al [129] proposed to use fitted qualitative shape labels for the purpose of generating synthetic abstract art from photographs.

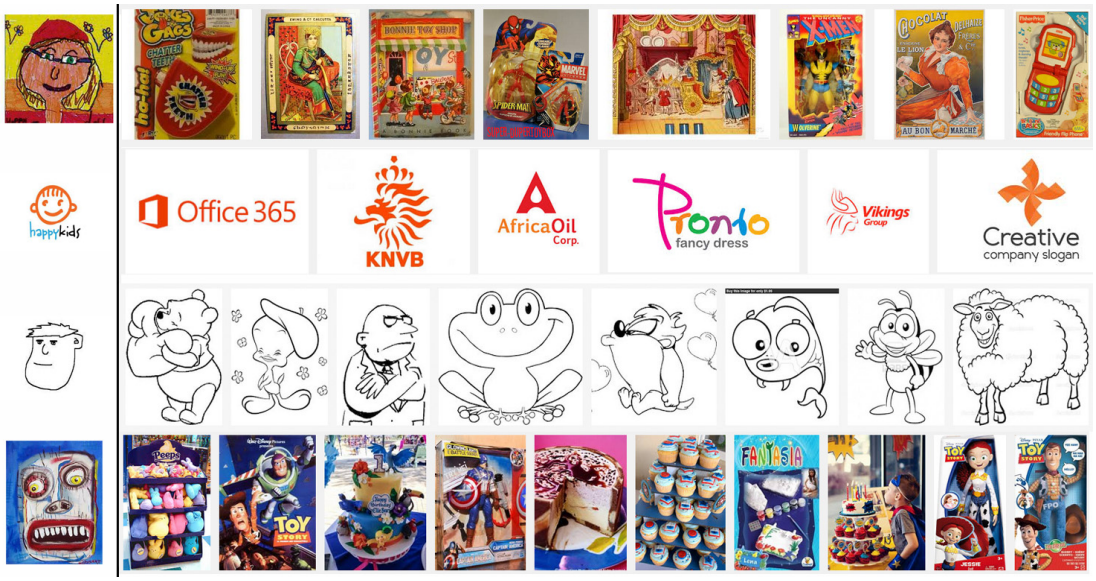


Figure 1-6: *Google image retrieval examples. Query images are displayed in the first column, the retrieval results returned from Google are shown on the right side.*

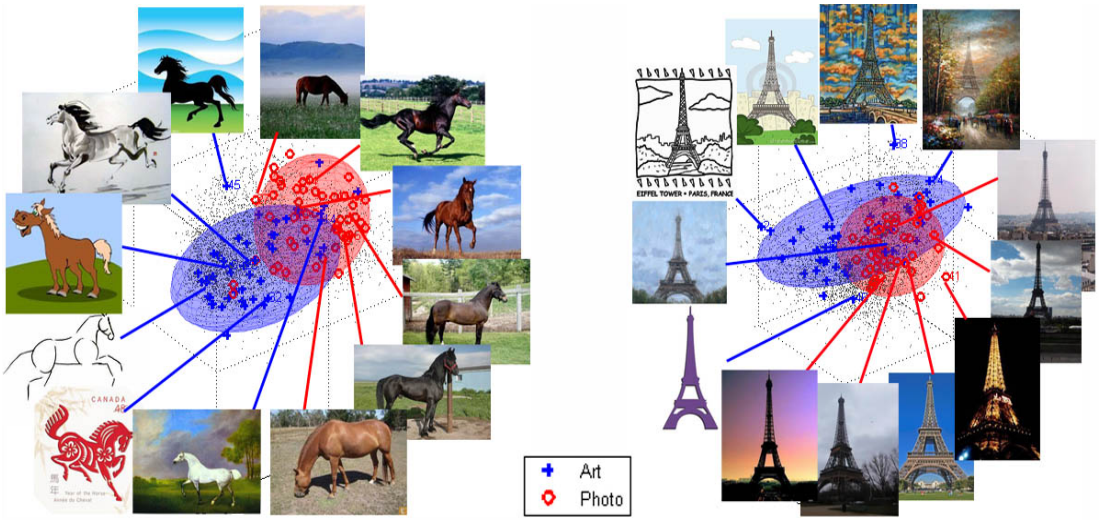


Figure 1-7: *Example images and their distribution in the feature space. The features are generated by projecting the 5000-d BoW-SIFT features to 3-d space using PCA. Gray clouds represent all categories in our* Photo-Art-50 *dataset. In the feature space, the art domain (blue) and photo domain (red) of the same object class distribute differently, partially overlapped. The art features tend to spread wider than the photo features, which is consistent with its higher variation of visual appearance.*

1.2 Challenges

Based on the motivations listed above, one can see that generalise to across depictions in Computer Vision applications is of importance, in both scientific and engineered view. Meanwhile, it is also a challenge one. Challenges mainly come from two folds. At first, the problem itself is hard - due to the wide variation of local features. Secondly, the dataset is hard to establish.

1.2.1 Wider Variation

Within our test, even the best classification and detection algorithms exhibit a significant drop in performance when presented with images that are not photographic (see section 5.2). Figure 1-7 provides a hint of the reason for such a performance drop. It shows the distribution of visual features for two specific classes. It shows the separation of objects in the same class but different depictions can be less than the separation between objects in different classes but the same depiction. This wide variation is a property of all visual classes we have tested, and underpins the intuition that the underlying difficulty in the cross-depiction problem is the seemingly unbounded number of distinct depictive styles. We also calculate and compare the K-L divergency (more details can be found in Chapter 5) of inter-depiction and inter-category for this two specific category. Inter-depiction divergencies are calculated based on the same class but depicted in different styles (such as photo and art) while the inter-category diver-

	Photo-Horse	Art-Eiffel
Photo-Eiffel	1.42	2.75
Art-Horse	2.33	2.15

Table 1.1: *Inter-depiction divergencies (in green cell) and inter-category divergencies (in blue cell). The K-L divergencies are calculated based on 5000-d BoW-SIFT features of 'Horse' class and 'Eiffel-Tower' class from Photo-Art-50 dataset.*

gencies are measured based on the same depicted style but from different categories. The divergencies table shows in table 1.1. From the table, it is shown that inter-depiction divergencies (Photo-Horse VS Art-Horse and Photo-Eiffel VS Art-Eiffel)are bigger than intra-category divergencies(Photo-Horse VS Photo Eiffel and Art-Horse VS Art Eiffel), which means the depiction variation is a much bigger challenge than the category variation usually considered in Computer Vision.

1.2.2 Dataset

Lacking of appropriate datasets and baselines makes our task much harder. There are many datasets for object detection, classification and recognition. Caltech-101 [45] is the first general objects dataset for classification. It contains images of 101 categories of object, and is relatively widely used within the community for evaluation object recognition. Caltech-256 [64] is build based on Caltech-101. It has 256 classes and there are at least 80 images in each class. PASCAL VOC [44] starts from 2005, updating every year, which is a dataset for classification, detection and segmentation. The latest VOC dataset is VOC 2012, which contains 20 classes and more than 25000 objects. ImageNet [37] is a much larger dataset, containing 200 classes with nearly 500000 images. However, all these popular datasets are built based on photorealistic images. Very rare artworks can be found in them. Doubtless, a dataset with artworks (such as painting, drawing, cartoon) will benefit the community. And it is important to the community to understand the performance of leading techniques in the context of cross-depiction problem.

We believe the solutions of these challenges would be of genuine benefit to computer vision. Advancing this area would provide a significant boost to current applications such as image search over a database. For example, given a photograph of the Queen of England, a search should return all portraits of her, ideally including the postage stamps that capture her likeness in bas-relief. More importantly, a solution to the cross-depiction problem forces us to consider ways to represent objects that are more general than appearance-based approaches currently used, which is also the main motivation of this thesis.

(a) (b) (c)

Figure 1-8: *(a) Picasso, Seated Woman with Wrist Watch. (b)Leger, Card Players. (c)Picasso, Three Musicians*

1.3 Our Contributions

Above sections highlight *what* problems we want to study in this thesis and *why* they are important. The next step is to identify *how* we approach the solutions, in other words, the technical contribution we made in this thesis. Our journey starts from exploring the common properties sharing between photos and art works. Shape is a good start since artists draw initial sketch using simple shapes to layout objects and scenes. The structure of the object remains relatively invariant to depiction in most object classes, and provides a description at a global level. After obtaining these weapons, a *hierarchical representation* of object class is built to classify objets depicted in different styles. To narrow the wide variation in visual appearance exhibited by visual objects across depictive styles, a *multi-labled graph representation* is then developed.

1.3.1 Finding Common Simple Shapes

Shape plays an important role in computer vision, with applications in problems such as matching, object recognition, and classification. However, to the best of our knowledge the question as to whether there is a set of elementary planar shapes that appear commonly in the world around us has never been asked within the literature. If such a set exists, then the elementary shapes could play a similar role in shape analysis as the primary colours do in colour analysis.

Our hypothesis, that images comprise combinations of primitive shapes, has its roots in observation, in art, and in psychology. Observation suggests the visual world can be described as an assembly of simple shapes: circles for wheels or faces, rectangles for cars, windows, human torsos, triangles for eyes, cats ears, mountains. These and similar primitives have been the basis of significant movements in 20^{th} century Western Art. Painters such as Picasso (*e.g.* Seated Woman with Wrist Watch), Leger (*e.g.* Card Players) shown in figure 1-8, and schools such as Italian Futurism, Tubism, and Orphism, depicted objects (and motions) as being composed of just a few basic

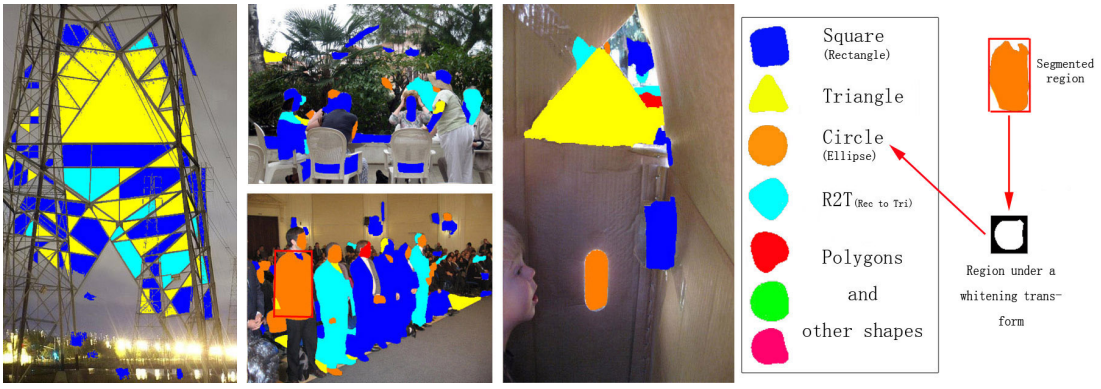


Figure 1-9: *Segmented regions classified by primitive shapes obtained from MIT database.*

geometric forms: cones, cylinders, bricks, and so on. It is very common for artists to make initial sketches using simple shapes to layout a scene, as any book on drawing instruction will testify.

Despite the anecdotal nature of this evidence, the practice is widespread enough and useful enough to suggest primitive shapes do regularly appear in real world images. Empirical evidence that aligns with artistic intuition has existed since at least the 1970s, when psychologists such as Rosch [113] showed simple shapes (specifically triangles, squares, and circles) are easier for humans to recall other shapes. Psychologists have explicitly used shape as a primitive to explain cognition in the form of Geons, a concept which comes from Biederman's theory [13]. Geons are the simple 2D or 3D forms such as cylinders, bricks, wedges, cones, circles and rectangles corresponding to the simple parts of an object of object recognition.

We describe an experiment designed to test the following hypothesis: *some regions in image segmentations can be classified and fitted as one of a few primitive shapes.* Not wishing to force regions into classes, we developed a classifier (with an input bandwidth) designed to find clusters of a size greater than would be expected if the shape of regions were randomly generated. We used two different "shape spaces" (*i.e.* shape descriptors), three different segmentation methods, and three image databases. The result was that primitive shapes (up to an affine transform) such as 'triangle', 'square', and 'circle' account for between 50% and 80% of regions. As Figure 1-9 shows, we can now classify image regions into qualitative shape classes (any region that touched a picture boundary or which contained less than 100 pixels was removed from consideration).

Our aim is to investigate whether primitive shapes exist in real world photographs, and more particularly whether shape classes could be discovered with as less human direction as possible. If this set exists, it can be used as a bridge between arts and photos. More details of this work can be found in chapter 3.

(a) A Healthy Breakfast

(b) A Happy Looking Healthy Breakfast!

Figure 1-10: *Spatial organisation of object parts plays an important role in the recognition of objects. In the above example, three seemingly unrelated parts may be combined to form a face. Humans are able to recognise the eggs as eyes, the strawberry as a nose and the sausage as a mouth only when they are spatially arranged akin to a human face. [6]*

1.3.2 A Hierarchical Structure as a Global Invariant

Structure of the object is another common properties share between photos and artworks. Wikipedia defines structure as

> " a fundamental, tangible or intangible notion referring to the recognition, observation, nature, and permanence of patterns and relationships of entities. "

An object's structure can be defined as the topology of its parts, with emphasis on the relations between the object and its parts, and also those between the parts themselves. The importance of structure has been highlighted by its use as a basis for representation of objects [1, 46, 54, 174, 173], and has been used for matching and detection.

An important hypothesis is that structure is class invariant and play a significant, possibly even essential role in modelling objects across depictions. Biederman [13] also emphasised that the spatial organisation of parts is important for recognising any objects. Balikai presented a good example to show the importance of structure played in object recognition in [6]. Consider the example shown in 1-10. In Figure 1-10(a), one may not relate the three objects to form a single object, but when the element of structure is introduced in Figure 1-10(b), humans can clearly relate the spatial arrangement to a face. This simple example underlines the vital role that structure plays towards recognition and understanding of objects invariant of their depiction [6].

In this thesis, we investigate a method for modelling visual objects classes in a manner that is invariant to depictive style. The assumption we make is that an object class is characterised by the qualitative shape of object parts and their structural arrangement. Hence we use a graph of nodes and arcs in which primitive shapes such as triangle, square, and circle to label the nodes. More exactly our model is a hierarchy of levels, yielding a coarse-to-fine representation. Each level contains an undirected graph of nodes and arcs. Nodes between levels are connected via parent-child arcs, which are directed. Child nodes are nested inside their parent. We also use our model on a cross-depiction image dataset. The experiments provide empirical evidence that our model is more robust to cross-depiction object classification than an excellent Bag of Words classifier. More details can be found in chapter 4.

However, structure may not exist in every category of object, for example, fluid like water and smoke. Moreover, some structural information might vary a lot even they are extracted from the same object category, such as buildings. The structure of different buildings differs a lot with the changing of building genre, especially for the modern buildings. In our research, we try to avoid including these object categories in our dataset since they are out of scope of our research at this stage. After all, the problem we interested in is how the depiction of object class affect the object recognition.

1.3.3 Multi-labeled Graph with Weights.

As we claimed in section 1.2, a challenge we want to address in this thesis is: how to capture the wide variation in visual appearance exhibited by visual objects across depictive styles. This variation is typically much wider than the lighting and viewpoint variations usually considered for photographic images. Indeed, if we consider different ways to depict an object (or parts of an object) there is a good reason to suppose that the distribution of corresponding features form distinct clusters. Its effect can be seen in Figure 1-11 where the currently accepted state-of-art method for object detection fails when presented with artwork. The same figure highlights our contribution by showing our proposal is able to locate (and classify) objects regardless of their depictive style.

We solve the problem of inter-depictive variation by using *multi-labeled* nodes to

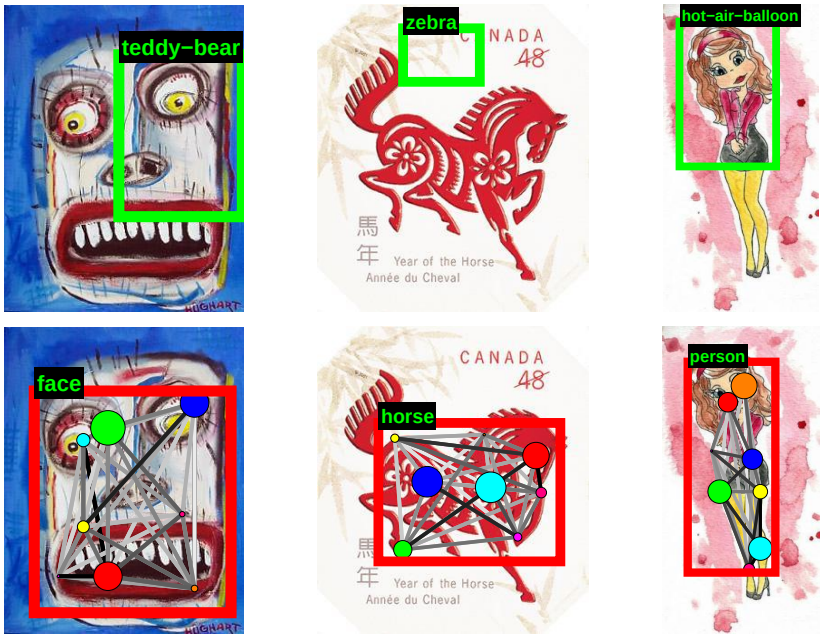


Figure 1-11: *Learning a model to recognise objects. Our proposed multi-labeled graph modelling method shows significant improvement for recognising objects depicted in variety styles. The green boxes are estimated by using DPM [46], the red are predicted from our system. The text above the bounding box displays the predicted class category over a 50-classes dataset. In our each detected window, the object is matched with the learned model graph. In the matched graph, each node indicates a part of the object, and larger circles represent greater importance of a node, and darker lines denote stronger relationships.*

describe objects parts. These multiple attributes are learned from different depictive styles of images, which are more effective than attempting to characterize all attributes in a monolithic model.

Moreover, in our model, a weight vector is learned automatically to encode the importance of node and edge similarity. We refer to it as the *discriminative weight* formulation for a part based model. This advantage will be demonstrated with evidence in the experimental section. More details of this work can be found in chapter 5.

1.3.4 Summary of Contributions

The main technical contributions of this thesis are:

1. Providing empirical evidence that some regions in segmented images can be classified or fitted as one of a few primitive shapes, upon given appropriate region descriptions and well-designed classifiers.

2. A classifier to fit primitive shapes to segmented regions of an object.

3. A computationally efficient classifier to categorise scenes based on ratio of primitive shapes.

4. A method for modelling visual objects classes in a manner that is invariant to depictive styles, using a hierarchical representation at global level with primitive shapes labeled in local level.

5. A modelling scheme (a framework) for visual class objects that generalise across a broader collection of depictive styles, using a novel weighted multi-labeled graph model.

Other contributions, includes:

1. A new agglomerative clustering method to cluster similar binary shapes.

2. A new challenge cross-depiction object dataset, *Photo-Art-50*, consisting of 50 classes, annotated with bounding boxes, designed specifically for the cross-domain problem.

3. An evaluation of leading recognition and detection techniques and two state-of-the-art domain adaptive methods for cross-depiction task.

1.4 A Road Map

Chapter 2 outlines the relevant background. It first provides a history and overview of the state of the art in object recognition, showing that our problem is hardly studied. We then reviewed the studies of shape, showing no one asked a question as we do – '*whether common simple shapes exist in natural images*'. After that, structural/graph based modelling methods are reviewed, to compare with our novel methods designed for cross-depiction problem. Finally, a couple of methods that study the problem of depiction-invariant modelling method are reviewed in details.

Regions segmented from natural images can be classified into a collection of primitive shapes (such as square, triangle, circle etc.). The experimental framework and method with experimental results are presented in chapter 3. Additionally, a primitive shape classifier and an application of scene classification application based on primitive shapes is also reported in this Chapter. In appendix A, some further experiments into the nature of shape description are carried out to show that the choices we make have little impact on the conclusion that primitive shape classes do exist.

Chapter 4 includes two sections. One explains how to build a hierarchical graph model to represent object classes, with nodes labelled by primitive shapes and edges labelled with displacement vectors. The other section describes experiments on a cross-depiction image dataset. The experiments provide empirical evidence that our model is more robust to cross-depiction object classification than an excellent Bag of Words classifier.

In the chapter 5, a new challenge cross-depiction dataset is presented and a baseline of leading recognition and detection techniques and two state-of-the-art domain adaptive methods for cross-depiction task are provided. Then, we describe our novel modelling scheme, and in particular introduces the way in which we account for the wide variation in feature distributions. A visual class model (VCM) is now a graph with *multi-labeled nodes* and *learned weights*. Such novel visual class models can be learned from examples via an efficient algorithm we have designed, and experimentally are shown to outperform state-of-art deformable part models at detection tasks, and state-of-art BoW methods for classification. All the experiments and results are provided in this chapter.

The thesis concludes, in chapter 6, with a discussion and observation drawn from experimental results. An overview of the possible future development and applications is also pointed in this chapter.

CHAPTER 2

LITERATURE REVIEW

Before reviewing related works, we want to first locate our literatures position in the subject of object recognition. Object recognition is one of the most fundamental problems in computer vision as well as the most challenge one. The challenges can be roughly divided to three levels, which are instance level, category level and semantic level [74], as shown in figure 2-1.

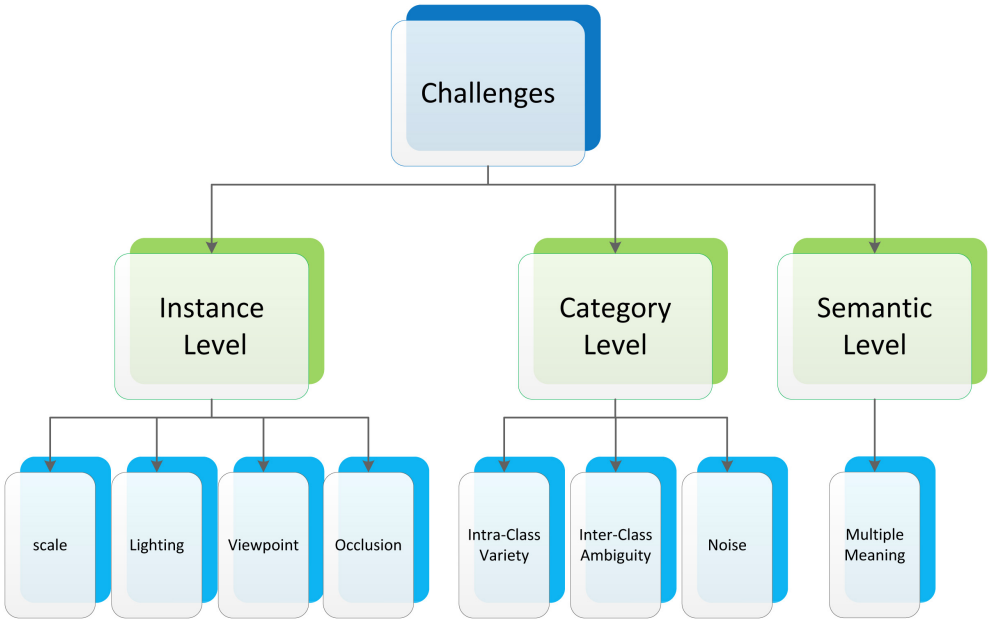


Figure 2-1: *Challenges of object detection and classification in different level. The depiction variation we want to address in this thesis is an important challenge belongs to the intra-class diversity under the category level.*

Instance Level: Challenges from this level usually come from the different capture conditions. Significant effort has been paid to develop modelling schemes and algorithms aiming at recognising generic objects in images taken under different conditions.

Figure 2-2: *Different examples of chairs*

For example, many approaches have been implemented to handle the fact that the image of the objects may vary somewhat in different view points [9, 46, 119, 138, 139], in many different scales or even when they are translated or rotated [48, 93]. Some researches focus on the problem when objects are partially obstructed from view [47, 161, 170].

Category Level: There are major three types of challenges at this level. At first, the intra-class diversity, it is caused by different visual appearance of the same object category. For example, chairs in figure 2-2, the visual appearance are so different. Multi-components models [46] and sub-categories models [39] are designed to address this problem. The depiction variation we want to address in this thesis is an important challenge belongs to the intra-class diversity under this level. The second challenge is called inter-class ambiguity, which is caused by the similarity between different object classes, such as a wolf and a Husky. Deep learning models such as [84] and [173] shows robustness in such kind of problems. The last challenge comes from the noise of background. In practical situation, the background could be very complicated and this will make the recognition much more challenging.

Semantic Level: This might be the hardest problem for object recognition - same image with multiple meanings. For example, the left image in figure 2-3 can be recognised as two against faces XOR a candle. The right image then can be explained as a head of a duck XOR a rabbit. Different explanations may relate to the observer's personality and experience, which is the hardest part for computer vision.

Figure 2-3: *Two examples of ambiguous image. Left: face or candle? Right: duck or rabbit?*

Object recognition has been studied for more than five decades [95, 144]. Many

researches, publications and applications are proposed to face and address the above challenges. However, a relatively less addressed issue is that of recognising objects regardless of their depiction.

Modelling visual object classes is an important step of relevance to object detection and classification. In this chapter, we first take a look at a few state-of-art modelling methods that is relevant to our work, including famous Bag of Words family to recently popular deep learning methods.

As a part of the work presented in this thesis, we want to investigate the common properties share between different depictions so that we can use them to capture the wide variations across different depictive styles. Shape and structure are what we have chosen and they have been applied individually in the past for modelling object category. Literature relating to the above properties have been reviewed in this chapter. We first look into prior works that relate to the shape study. Then, the following section reviewed works that use structural information to model object class. Finally, a couple of methods that study the problem of depiction-invariant modelling method are reviewed in details.

2.1 State-of-art in Modelling Object Class

How to represent a visual object class is the key to detect and classify such an object in an input image. For object classification, it is the task of finding whether a sort of object exists in the image. Generally, algorithms of object classification describe entire image through hand-crafted (such as HoG, LBP, SIFT) or self-learned features (such as Caffe features) at first. Then, a classifier will be learned to estimate what sort of object class features exist in those image features, in order to classify the image. Object detection is much more complicated, which needs to answer the question of 'what object is in where'. Hence, except for the local features, object structural information is also an important aspect for detection. In recent years, most researches of modelling object class for classification is focused on learning feature representations, such as Bag-of-Words models, deep learning models; while object detection is more focused on structural learning, such as the pictorial structures framework. The family of these state-of-art methods will be reviewed in the following sub-sections.

2.1.1 Bag-of-Words

The bag-of-words model (BoW) is initially used in natural language processing and information retrieval. It is commonly used in methods of document classification, where the frequency of occurrence of each word is used as a feature for training a classifier. In 2004, it was adapted for computer vision applications by Csurka et al [33]. To represent an image using BoW model, an image can be treated as a document and

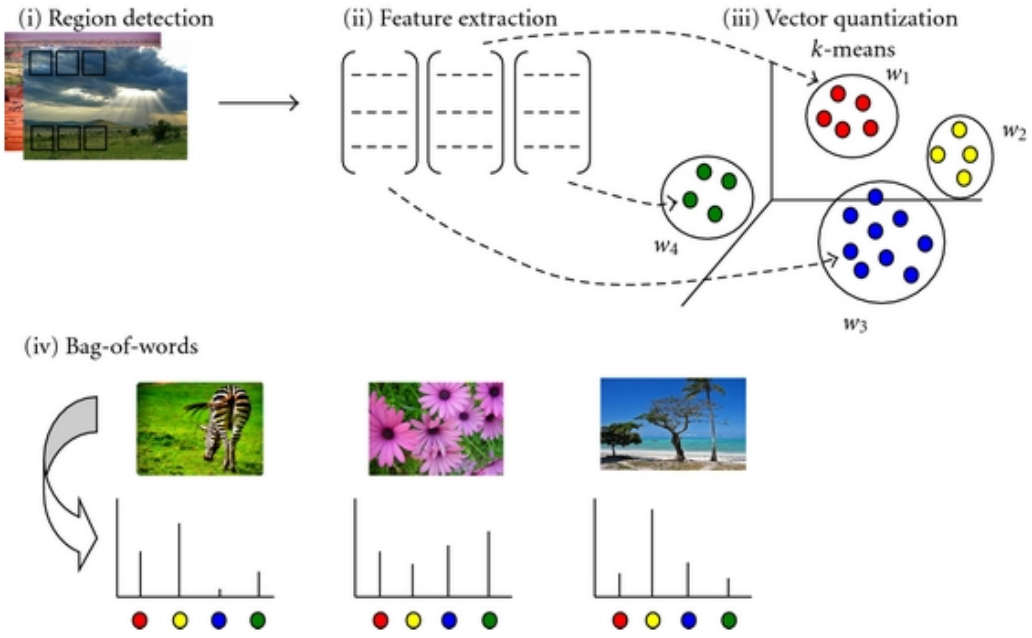


Figure 2-4: *Constructing the bag-of-words for image representation.[141]*

a visual analogue of a word is used. It is based on the vector quantization process by clustering low-level visual features (such as color, texture etc) of local points or regions. Four steps are included to represent image using BoW model, which are feature extraction, feature description, codebook generation and model learning. Figure 2-4 describes these steps to construct BoW model from images. We review works related to these steps, separately, in the following paragraphs.

1). Feature Detection: There are two category methods to extract features, sparse based and dense based. For those feature extraction of sparse interest point, they are computed at pixels, edges, corners, blobs and so on. Most commonly used sparse feature detectors include Harris corner detector [68], FAST(Features from Accelerated Segment Test) corner detector [114], LoG(Laplacian of Gaussian), DoG(Difference of Gaussian) [92] and MSER (Maximally Stable Extremal Regions) [97] etc. Sparse based detector is efficient since only parts of the image features are detected and selected. In recent years, with the increase of CPU computation capability, dense based feature extractors become more and more popular. They extract large scale image features on a dense grid of locations at a fixed scale and orientation. Dense based methods can obtain much more information, although with high redundancy. However, with better feature representations and encoding methods, dense descriptors have been proved to perform better than sparse based ones [14, 81].

2). Feature Description: SIFT(Scale Invariant Feature Transform) descriptor [92] is the most widely used local descriptor. It combines a scale invariant region detector and

a descriptor based on the gradient distribution in the detected regions. A SIFT descriptor is a 3-D spatial histogram of the image gradients in characterizing the appearance of a key point. The gradient at each pixel is regarded as a sample of a three-dimensional elementary feature vector, formed by the pixel location and the gradient orientation. Samples are then weighed by the gradient norm and accumulated in a 3-D histogram, which (up to normalization and clamping) forms the SIFT descriptor of the region. An additional Gaussian weighting function is applied to give less importance to gradients farther away from the key point centre. In most situation, the frame will be divided into four by four grids, in each grid, there are 8 directions. So there will be 128 dimensions. Some other descriptors include HOG(Histogram of Oriented Gradient) [34], LBP(Local Binary Pattern) [103] etc.

3). Codebook Generation: The next step for the BoW model is to convert vector represented patches to 'codewords', which also produce a 'codebook'. In general, the k-means clustering algorithm is used for this task at first and then codewords are defined as the centers of the learned clusters and the number of visual words generated is based on the number of clusters. Hence, each patch in an image is mapped to a certain codeword through the clustering process and the image can be represented by the histogram of the codewords. This is known as vector quantization coding process and which is the most commonly used one. One drawback of this codebook approach is the hard assignment of codewords in the vocabulary to image feature vectors. The hard assignment gives rise to two issues: codeword uncertainty and codeword plausibility. Codeword uncertainty refers to the problem of selecting the correct codeword out of two or more relevant candidates. The codebook approach merely selects the best representing codeword, ignoring the relevance of other candidates. The second drawback, codeword plausibility denotes the problem of selecting a codeword without a suitable candidate in the vocabulary. The codebook approach assigns the best fitting codeword, regardless the fact that this codeword is not a proper representative. Gemert et al [145] propose an uncertainty modeling method for the codebook approach. In effect, they apply techniques from kernel density estimation to allow a degree of ambiguity in assigning codewords to image features. Some other feature encoding algorithms include sparse coding [105], locality-constraint linear coding[149], salient coding [73], fisher vector coding [107], super vector coding [172] and so on. The super vector coding [172] and fisher vector coding [107] are the best performance coding methods in recent years. They are very similar since they are coding the difference between local features and visual words. For fisher vector, not like the classical coding ways, it records both the first order and second order difference. And for super vector, it directly uses the difference between local features and its nearest visual word to instead the previous hard assignment.

To capture the spatial information in order to improve the limitations of the con-

ventional BoW model, many studies have been proposed, in which spatial pyramid matching introduced by Lazebnik et al. [86] has been widely compared as one of the baselines. This technique works by partitioning the images into increasingly fine sub-regions and computing histograms of local features inside each sub-region.

4). Model Learning: After the BoW feature is extracted from images, it is entered into a classifier for training or testing. The most widely used and developed classifier is based on support vector machines (SVM) [146].

2.1.2 Deep Learning Models

Deep learning models [11] aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. The mainstream deep learning models include Auto-encoder [68], RBM(Restricted Boltzmann Machine) [127], DBN(Deep Belief Nets) [69], CNN(Convolutional Neural Networks) [87] etc.

Auto-encoder[68] is an artificial neural network proposed in 80s of last century, which is widely used in dimensionality reduction and feature extraction. Auto-encoder is composed by an *encoder* and a *decoder*. Encoder transforms the input to hidden layer so that the representation can be reconstructed by the decoder. In the process, the hidden units learn to project the input in the span of the first several principal components of the data, achieving the dimensionality reduction and feature encoding. Auto-encoder has achieved good performance in handwriting recognition and image classification.

RBM [127] is an undirected bipartite graph model, which is a typical Enery-based Model (EBM). As its name implies, RBM is a variant of Boltzmann Machine, with the restriction that their neurons must form a bipartite graph. With that special structure, a very efficient Gibbs sampling can be performed to obtain an estimator of the log-likelihood gradient. RBM can be used as an unsupervised feature learning unit.

Hinton et al in the University of Toronto introduced Deep Belief Networks (DBNs) [69], with a learning algorithm that greedy trains one layer at a time, exploiting an unsupervised learning algorithm for each layer, a Restricted Boltzmann Machine. The multi-layers architecture of DBN makes it possible to learn a hierarchical feature representation to achieve feature auto-encoding. DBN has been successfully employed in handwriting recognition, speech recognition, content based image retrieval and so on.

Convolutional Neural Networks might be the most widely used models for image recognition. They are inspired by the visual system's structure, and in particular by the models proposed by [75]. The first computational models are found in Fukushima's Neocognitron [57], which are based on local conductivities between neurons and hierarchically organized transformations of the image. He recognized that when neurons with the same parameters are applied on patches of the previous layer at different locations,

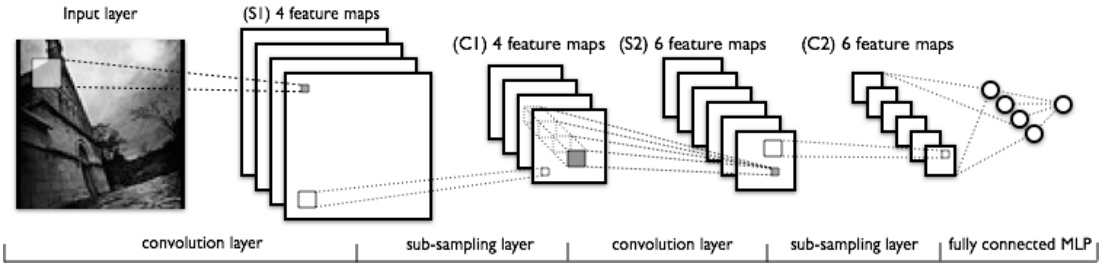


Figure 2-5: *A graphical depiction of a LeNet model. The lower-layers are composed to alternating convolution and max-pooling layers. The upper-layers however are fully-connected and correspond to a traditional MLP (hidden layer + logistic regression). The input to the first fully-connected layer is the set of all features maps [87].*

a form of translational invariance is obtained. Later, LeCun and his collaborators, following up on this idea, designed and trained convolutional networks using the error gradient, obtaining state-of-the-art performance [87] on several pattern recognition tasks. Figure 2-5 shows a graphical description of a leNet model. And more recently, works based on these networks have achieved competition-winning numbers on large benchmark datasets consisting of more than one million images, for image recognition task.

2.1.3 Deformable Models

Deformable models of various types are widely used to model the object class for detection tasks. On difficult datasets, deformable models are often outperformed by simpler models such as rigid templates or bag-of-words. There is a significant body of work on deformable models, including several kinds of deformable template models [28, 29] and a variety of part-based models [3, 30, 47, 46, 48, 55, 88].

In the constellation models from [48], parts are constrained to be in a sparse set of locations, and their geometric arrangement is captured by a Gaussian distribution. In contrast, pictorial structure models [47], originally introduced by Fischlet and Elschlarger [55], provide a statistical model of objects. The basic idea is to represent an object by a collection of parts arranged in a deformable configuration. The appearance of each part is modeled separately, and the deformable configuration is represented by spring-like connections between pairs of parts. Using these pictorial structure models, objects in an image can be recognized and their constituent parts can be located in the image. Figure 2-6 shows a pictorial structural representation of human face, indicating parts and their linkages. The patchwork of parts model from [3] is similar, but it explicitly considers how the appearance model of overlapping parts interact.

Deformable Part-based Model (DPM) [46] is the most successful one in the deformable models family and it is largely based on the pictorial structures framework from [55]. They use a dense set of possible positions and scales in an image, and define

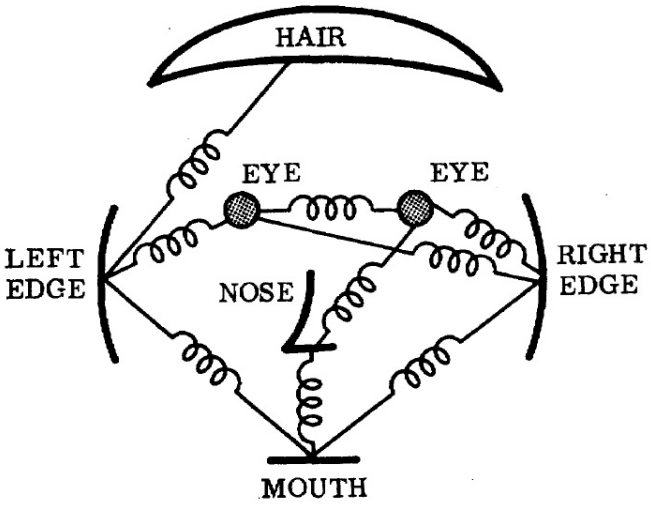


Figure 2-6: *A pictorial structural representation of human face, indicating parts and their linkages.[55]*

a score for placing a filter at each of these locations. The geometric configuration of the filters is captured by a set of deformation of costs connecting each part filter to the root filter, leading to a star-structured pictorial structure model. We will introduce this model in more details in section 2.3, where proves that structure is an important property to model object class across depictive styles.

2.1.4 Discussion

In above sections, we introduced some famous state-of-art for modelling object classes. Some of them are based on the learning of feature representations (such as BoW and Deep Learning) and some are based on deformable structures (such as DPM).

Although the BoW methods address many difficult issues, they tend to generalise poorly across depictive styles. The explanation for this is the formation of visual codewords in which clustering assumes low variation in feature appearance. To overcome this drawback, researchers use alternative low-level features that do not depend on photometric appearance, *e.g.*, edgelets [123, 51] and region shapes [66, 76]. However, even these methods do not generalise well.

For deep learning models, although they have been studied for nearly 30 years, they are not widely used in object classification until very recently. To the best of our knowledge, there is no systematic study to examine the performance of deep learning models on a cross-depiction dataset, although it has been proved that they can be used in domain adaption problems. Notice that the 'domain' here only means photos took under different photographic conditions, which is not comparable with our photo-art domain.

Deformable models, by modeling objects from different views with distinct models,

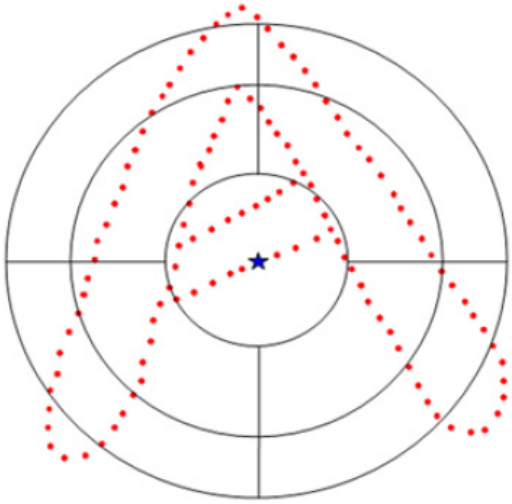


Figure 2-7: *An example of shape description using shape context [10]. This figure illustrate the extraction of shape context using log polar bins for the contour point highlighted by the star.*

it is able to detect large variations in pose. However, when the variance comes from local parts, *e.g.* the same object depicted in different styles, it does not generalize well; this is exactly the problem we address.

We argue that no single 'monolithic' feature will cover all possible appearances of an object (or part), when depictive styles are considered. We commit ourselves to find out the common properties of objects in photos and art works, and then we find an appropriate way to model the object class across different depictive styles. In the following sections, we will review some works related with 'shapes' and 'structures', which are the shared properties between photos and arts.

2.2 Shapes

The literature studying planar shape is large and it covers many areas. Within image processing and computer vision the shape literature is large and growing larger. Shape representation is of use to many applications. Our interest in the subject is classification using quantitative terms. Most of the literature develops quantitative measures, so we will provide a brief, targeted background.

Boundary based descriptors permit a scale based representation [101]. There are many features that depend on boundary descriptors of objects such as blending energy, curvature etc. For an irregularly shaped object, the boundary is a better representation although it is not directly used for shape descriptions like centroid, orientation, area etc. Fourier descriptors are also a common example [108, 115, 110]. It measures the regularity of a shape by analyzing its radial distance. Some other earlier works for shape description are based on silhouettes, such as [100] and [120]. [100] is based on

finding points of inflection on the curve at varying levels of details using path length parameter and combining them to obtain the scale space image of the curve. [120] propose that a symmetry-based representation is an intermediate representation that retains the advantages of local, edge-based correlation approaches as well as of global, deformable models.

However, as Belongie and Malik noted in [10], silhouettes are limited because they ignore internal contours. Hence, some works represent shapes as loose collections of 2D points (such as [27, 58]) or other 2D features ([42] and [50]). Other works propose more informative structures than individual points as features, in order to simplify matching. Belongie and Malik [10] propose the Shape Context, which has been populated in several applications to find similarities between corresponding points in a pair of images. Iteratively, every point in the image is used as a reference to build a log polar distribution of all other points on the shape, giving rise to a point-wise feature vector termed as shape context. Figure 2-7 shows an example. This shape description enables the point-to-point matching between two shapes even under non-rigid deformations. In effect, matching two shapes using shape context aligns them with each other, making the method invariant to changes in pose.

Leordeanu et al. [89] encode relations between all pairs of edgels of shape to go beyond individual edgels. Similarly, Elidan et al. [42] use pairwise spatial relations between landmark points. Ferrari et al. [50] propose a family of scale invariant local shape features formed by short chains of connected contour segments.

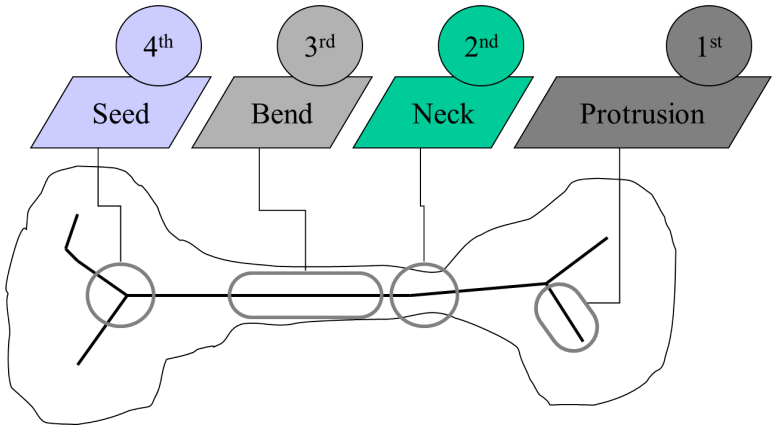


Figure 2-8: *An example showing a shock graph as a combination of singularities obtained during the evolution of the grassfire transform[125]*

Shape skeletons are the dual of shape boundary, and also have been used as a descriptor. For example, Rom and Medioni [112] suggest a hierarchical approach for shape description, combining local and global information, to obtain skeleton of shape. Sundat *et al* [131] use skeletal graph to represent shape and use graph matching techniques to match and compare skeletons. Shock graph [125] is derived from skeleton models

of shapes, and focus on the properties of the surrounding shape. Shock graphs are obtained as a combination of singularities that arise during the evolution of a grassfire transform on any given shape. In particular, the set of singularities consists of corners, lines, bridges and other similar features. Shock graphs are then organised into shock trees to provide a rich description of the shape. Figure 2-8 shows an example.

Region based descriptors are robust to noise when compared with either boundary or skeletal descriptors. In region-based methods, all pixels within a shape region are taken into account to obtain the shape representation. Common region-based methods use moment descriptors to describe shape. Typical descriptors include geometric moments such as Hu [70], Zernike [134], and Chebyshev [109]. In general, moments are constructed by using a set of complex polynomials which form a complete orthogonal basis set defined on the unit disk. It is a certain particular weighted average (moment) of the image pixels' intensities, or a function of such moments, usually chosen to have some attractive property or interpretation. Geometric moments representations interpret a normalized gray level image function as a probability density of a 2D random variable [171]. The first seven invariant moments, derived from the second and third order normalized central moments, are given by Hu [70]. There is no general rule in acquiring higher order invariants. And it has been shown [136] that Zernike moments outperform other moments in terms of noise sensitive, redundancy and reconstruction error. In [82], Zernike moments have been used for image retrieval and have shown good results. Alternative region based descriptors also exist, such as [56], which computes descriptor using the scale invariant feature transform (SIFT), with the resampled MSER binary mask as input.

Shape has been put to use in many computer vision tasks not limited to matching [10], classification [137], and retrieval [35]. Particularly, decomposing images into regions and shapes of geometric parts has gained popularity in recent years. Tu et al [143] define an image parsing framework by decomposing an image into its constituent visual patterns and it outputs a 'parsing graph' that can improve image segmentation on natural images of complex city scenes. In [67], Han and Zhu observe that many man-made scenes in natural images can be decomposed hierarchically into a small number of primitives arranged by a small set of spatial relations and they present a simple attribute graph grammar as a generative image presentation to improve the detection. However, they limited their fitted segmented regions to be rectangles and only man-made scenes, while we do not set such limitations. Teboul et al [135] address shape grammar parsing for facade segmentation using reinforcement learning and Riemenschneider et al [111] provide feasible generic facade reconstruction by combining low-level classifiers with high-level object detectors to infer an irregular lattice. In [43], Eslami et al use a type of Deep Boltzmann Machine that they call a Shape Boltzmann Machine (ShapeBM) for the task of modelling binary shape images. They show that the

ShapeBM characterizes a strong model of shape, in that samples from the model look realistic and it can generalize to generate samples that differ from training examples.

It is not possible, nor is it our purpose, to review the extensive shape related literature here; the above is a small but representative sample. What is important to this thesis is that none of the literature we know of asks as we do: *" is there a set of shapes commonly present in natural images?"* Intuitively we would expect such shapes to be simple, but most of the existing literature — especially recent publications — use relatively complex silhouettes of real objects (cups, horses, hands *etc*). This thesis tests the proposition that simple (primitive) shapes exist in natural images — that they are part of 'the signal'.

The study of simple nameable shapes has lead to some promising applications in representing object. For example, Balikai et al. [8] propose a method to describe any images using a collection of known shapes, specifically: ellipses, rectangles, triangles and convex hull. Similarly, Song et al. [129] show that classifying simple shapes is a tool useful in non-photorealistic rendering from photographs. The classifier inputs regions from segmentation and outputs the 'best' fitting simple shape such as circle, square, or triangle. Although these works use simple shapes (such as triangles, rectangles, circles etc.) as image features, the evidence of '*why use these simples shapes, but not other shapes?*' is lacking, too. The missing of these researches motivated us to find out whether common simple shapes objectively exist in natural images. If such a set exist, it could play as a common property in both photos and art works.

To apply above reviewed shape descriptors, one first needs to obtain binary shape of the depicted object. Segmenting the object into regions or salient parts is an option to achieve this. Moreover, this option leads to the idea using object structure as a global invariant for the differently depicted object class. The following section reviews the literature that using structure to model visual object classes.

2.3 Structures

An object's structure can be defined as the topology of its parts, with emphasis on the relations between the object and its parts, and also those between the parts themselves. The study of object structure actually starts from psychology, according to Gestalt's Laws [83, 150], the human brain groups parts of an object to recognise a holistic visualisation of the object. The parts are organised or grouped based on a set number of rules, which in turn depend on the intrinsic properties of the parts and their relationships. Rules include symmetry, similarity, proximity, closure and smoothness. In any situation, one or more of these rules may be used to semantically perceive an object. The fact that these rules are geometric and not particularly restricted in their application to any one depiction style, encourage computer vision scientists to use such

geometric rules for representing and interpreting objects.

In 1972, Biederman et al [13] stated that an object can be structured represented as a combination of a relatively small set of simple 2D and 3D shapes. The theory proposed that visual input is matched against structural representations of objects in human brain. In some cases, even the same parts with different structural arrangement may represent different object classes. For example, as Figure 2-9 shows, a cone on the top of an sphere can be recognised as a toy's head, conversely, a sphere on a cone is like a ice-core. Moreover, they presented that human are able to identify objects correctly

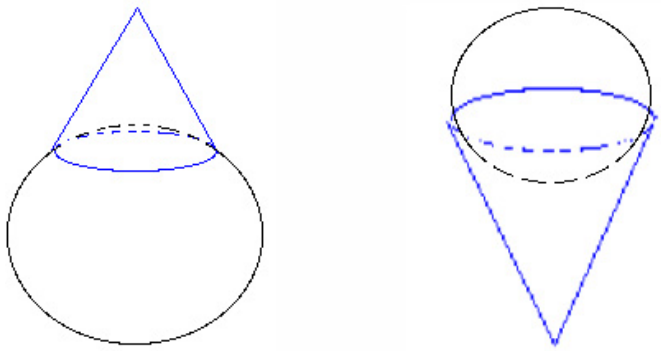


Figure 2-9: *Two cases of two interrelated geons, What does the reader imagine in each case?*

if a subset of only two or three components are available and they are in the correct spatial organization. Because of the importance of structural information to the object recognition for human, it has been employed in many object modelling methods.

The neuroscientist, psychologist and the founder of the computer vision, David Marr also stated that human interpret objects from 2D to 3D in three stages [95]:

- **Primal Sketch (2D)** is based on feature extraction of fundamental components of the scene, such as edges, regions, etc., similar to a pencil sketch drawn quickly by an artist as an impression.
- **2.5D Sketch** where textures are included, similar to when the artist highlights the sketch by shading or painting.
- **3D Model** where the object is visualised as a continuous 3D map.

We can see how these theories play a major role even when a child draw an object - they draw from what they have memorised but not they are seeing, because object structure has been carved in their mind. All of the above theories and examples highlight beyond doubt the vital and essential role played by the structure of object class. Then we will review some works in computer vision that use the object structural information as an object class invariant.

Structure can be extracted from shapes, some of works we reviewed in the previous section can be extended for the purpose of defining structure. We have already seen an

example of this in [125], where shock graphs are organised into shock trees, implicitly encapsulating the structure of the object being described. Similarly, shape context [10] also encapsulates structure by encoding relative spatial distributions of points on the contour of the shape.

A more popular approach to obtain a structural representation of an object is to break it into its salient parts and connect each other by edges. Pictorial structure models are first introduced by Fischler and Elschlager [55]. The basic idea is to represent an object by a collection of parts arranged in a deformable configuration. While the pictorial structure formulation is appealing in its simplicity and generality, several shortcomings have limited its use:(i) the resulting energy minimization problem is hard to solve efficiently, (ii) the model has many parameters, and (iii) it is often desirable to find more than a single best (minimum energy) match. In [47], Felzenszwalb addressed these limitations, providing techniques that are practical for a broad range of object recognition problems. He restrict the structural graph to be acyclic and the relationships between connected pairs of parts be expressed in a particular form.

In [30], Crandall et al introduced a class of graphs called k-fans. Graphical models defined by k-fans provide a natural family of spatial priors for part-based recognition. The parameter k controls both the representational power of the models and the computational cost of doing inference with them. At one extreme, $k = 0$, there is no dependence between the locations of different object parts. When $k = 1$ the structure is that of a star graph. By providing explicit control over the degree of spatial structure, the models make it possible to study the extent to which additional spatial constraints among parts are actually helpful in detection and localization, and to consider the tradeoff in representational power and computational cost.

Kumar et al [85] extend pictorial structures in a number of ways: in particular, both the outline and the enclosed texture of the part are included in its appearance parameters and all parts are connected to each other to form a complete graph instead of a tree structure. A properly normalized measure of the probability of a part being present at a location is modelled using the PDF projection theorem.

Deformable part-based model developed by Felzenszwalb et al [46] is also based on pictorial structure framework. In their proposed method, they use a star-structured part-based model defined by a "root" filter plus a set of parts filters and associated deformation models. The score of a star model at a particular position and scale within an image is the score of the root filter at the given location plus the sum over parts of the maximum, over placements of that part, of the part filter score on its location minus a deformation cost measuring the deviation of the part from its ideal location relative to the root. In DPMs, the part filters capture features at twice the spatial resolution relative to the features captured by the root filter. Hence, it is actually a hierarchical graph with two layers. To train models using partially labeled data they

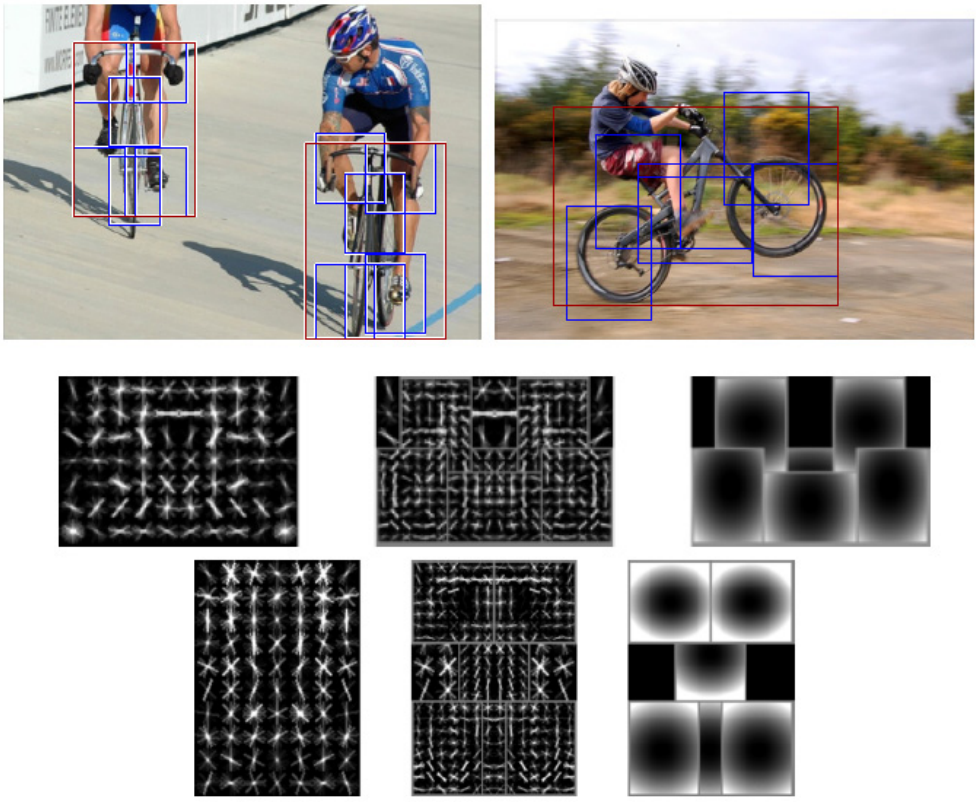


Figure 2-10: *Detections obtained with a 2 component bicycle model. In this model the first component captures sideways views of bicycles while the second component captures frontal and near frontal views [46].*

use the laten SVM (LSVM). In a latent SVM each example x is scored by a function of the following form,

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z) \tag{2.1}$$

Here β is the concatenation of the root filter, the part filters, and deformation cost weights, z is a specification of the object configuration, and $\Phi(x, z)$ is the concatenation of subwindows from a feature pyramid and part deformation features. Moreover, by modelling objects from different views with distinct models, it is able to detect large variations in pose. Figure 2-10 shows detections obtained with a 2 component mixture bicycle model.

Most recently, Lin et al [91] proposed a novel reconfigurable part-based model, namely And-Or graph model, to recognise object shapes in images. The proposed model consists of four layers: leaf-nodes at the bottom are local classifiers for detecting contour fragments; or-nodes above the leaf-nodes function as the switches to activate their child leaf-nodes, making the model reconfigurable during inference; and-nodes in a higher layer capture holistic shape deformations; one root-node on the top, which is also an or-node, activates one of its child and-nodes to deal with large global variations

(e.g. different poses and views). Figure 2-11 shows an example of And-Or graph model.

A tree structure also can be obtained from segmentations. Nodes at upper levels correspond to larger segments, while their children nodes capture embedded, smaller details. This tree is named as Segmentation Tree(ST) [1]. However, ST cannot distinguish many different ways in which the same set of subregions may be spatially distinguished within the parent region, giving rise to significantly different visual appearances, while their hierarchical properties remain fixed. Consequently, STs for many visually distinct objects are identical.

Connected Segmentation Tree (CST) [2] is an extension of ST to represent object using a hierarchical structural graph, which captures canonical characteristics of the object in terms of the photometric, geometric, and spatial adjacency and containment properties of its constituent image regions. CST is obtained by augmenting the object's segmentation tree (ST) with inter-region neighbor links, in addition to their recursive

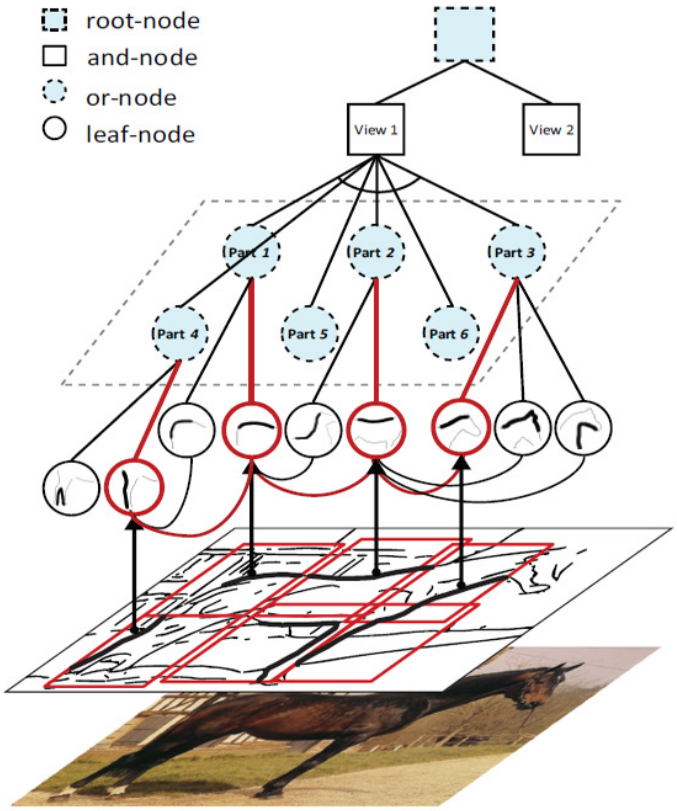


Figure 2-11: *It comprises four layers from bottom to top: the leaf-nodes (denoted by the solid circles) at the bottom for localizing local contour fragments, the or-nodes (denoted by the dashed blue circles) over the bottom specifying the activations of their child leaf-nodes, the and-nodes (denoted by the solid squares) encoding the holistic (view-based) variances, and the root-node (denoted by the dashed blue squares) on the top to switch the selection of its child and-nodes. The horizontal links incorporate contextual interactions among parts. Note that the leaf-nodes inherit the links that are defined between the layer of or-nodes. The nodes and links in red indicate the activation of leaf-nodes during the detection. [91]*

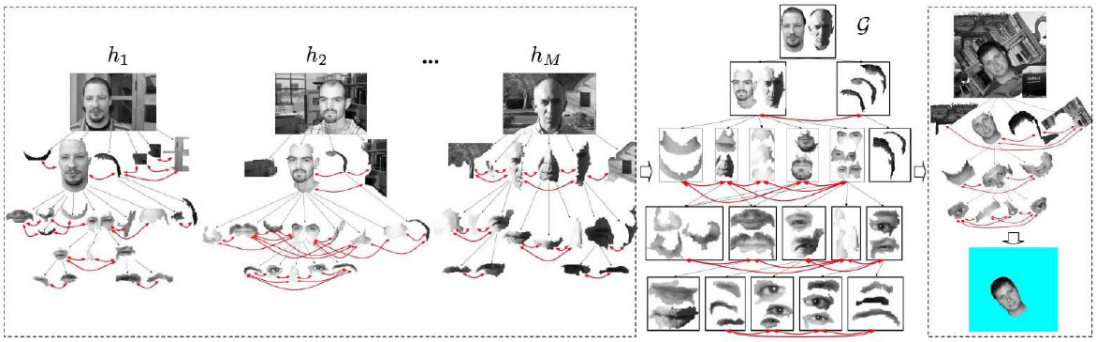


Figure 2-12: *Training images containing faces are represented by CSTs which capture the recursive containment (black edges) and neighbor relationships (red edges) of regions. Similar common subgraphs of the CSTs (faces), are registered and fused into the category model G. CST of a new image is matched with G to simultaneously detect, recognize, segment, and explain face occurrences. [2]*

embedding structure already present in ST. This makes CST a hierarchy of region adjacency graphs. A region's neighbors are computed using an extension to regions of the Voronoi diagram for point patterns. Unsupervised learning of the CST model of a category is formulated as matching the CST graph representations of unlabeled training images, and fusing their maximally matching subgraphs. Figure 2-12 shows a face model learning from a sequence images and detections in a test image.

Song et al [128] proposed a method based on Laplacian Graph Energy to identify semantic structures in image hierarchies. In this work, a segmentation tree (ST) is first built based on hierarchical global-Pb edge maps, then, they use component-wise Laplacian Graph Energy (cLGE) to filter out noisy levels in the hierarchy by removing levels that do not contribute much to the overall complexity of the graph and that do not change the overall meaning of the graph. The meaningful connections are left to construct the semantic structure. Two examples are shown in 2-13.

Structure can be used as a visual class invariant to depictive style is first proposed in [166]. In this paper, they believe the topology of an object's parts is a class property invariant to depiction. More specifically, a hierarchical structure of object is extracted at first. The topological relationship between parts is then characterized with a feature vector, using the normalized Laplacian function. In recent work, Balikai and Hall [7] proposed a more general approach for describing objects invariant to depictive style. They use structure at a global level, which is combined with a simple non-photometric descriptor(SSD) at a local level.

2.4 Cross-depiction Studies

In the above sections, we first review the state-of-art in the area of object recognition, including famous BoW and DPM. Then, important image descriptions such as shape

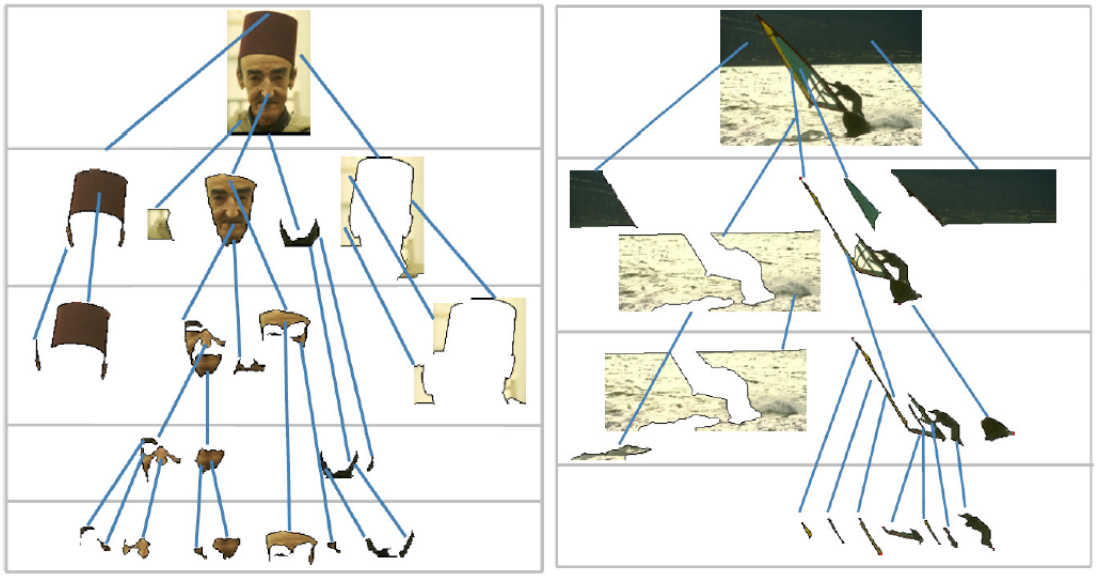


Figure 2-13: *How objects are broken into useful parts using Laplacian Graph Energy. [128]*

and structure are examined in a historical way. It is obvious that cross-depiction problems are comparably less well-explored, although there are hundreds of paper published in the area of object detection and classification every year. We review some important works directly related to cross-depiction problems as following, both for specific styles and general solutions.

2.4.1 Particular Styles

Some works only focus on particular styles, for example, Fidler et al [53] present an approach that enables unsupervised learning of generic parts of object structure within a hierarchical framework by exploiting the regularities present in the visual data. This proposed hierarchical learning framework has been applied on a clip-art dataset, which only covers limited depictive styles such as cartoons and line-drawings.

Sketch based image retrieval (SBIR) has attracted a lot of attentions. Bimbo and Pala [36] present a technique which based on elastic matching of sketched templates over edge maps in the image to evaluate similarity. The degree of matching achieved and the elastic deformation energy spent by the sketch to achieve such a match are used to derive a measure of similarity between the sketch and the images in the database and to rank images to be displayed. Chans et al. [23] tokenize edge segments into a string representation, encoding length, curvature, and relative spatial relationships. Chale et al. [22] employ angular-spatial distribution of pixels in the abstract images to extract features using the Fourier transform. The extracted features are rotation and scale invariant and robust against translation. Chen et al [25] propose Sketch2Photo - an interactive system in which keyword-annotated sketches are used to retrieve and

Figure 2-14: *Example query sketch, and their top ranking results (ranking from left to right) over the Flickr15K dataset. Results produced using GF-HOG descriptor in a Bag of Visual Words (BoVW) framework with vocabulary k = 3500 and histogram intersection distance.[72]*

composite photograph fragments. In that system, keywords trigger a Google Image search which returns possible images - the sketch is used only to crop the image using coarse shape matching via Shape Contexts. While in [19], major curves of images are first detected, based on which a curve-based algorithm is conducted to achieve precise matching between sketch and image database. In [18], Cao et al proposed a novel index structure and the corresponding raw contour-based matching algorithm to calculate the similarity between a sketch query and natural images, and make sketch-based image retrieval scalable to millions of images. Different from well-known bag-of-features representation in local feature-based image retrieval, where the visual vocabulary is quantized in the visual space, they describe a visual word using a triple (x, y, θ) of the position of an edge pixel (edgel) and the edgel orientation at that position. And a matching algorithm called structure-consistent sketch matching is proposed to measure the similarity between a sketch query and a database image.

Li et al [90] present a method for the representation and matching of sketches by exploiting not only local features but also global structures, through a star graph. Edge-based HoG was explored in [72] to retrieve photographs with a hand sketch query. In order to encode the relative location and spatial orientation of sketches or Canny edges of images, they represent image structure using a dense gradient field interpolated from the sparse set of edge pixels. Figure 2-14 shows some sketch based image retrieval examples using this method.

Other than sketches2photos, Russel et al [116] addresses the problem of automatically aligning historical architectural paintings with 3D models obtained using multi-view stereo technology from modern photographs. This is a challenging task because of the variations in appearance, geometry, color and texture due to environmental changes

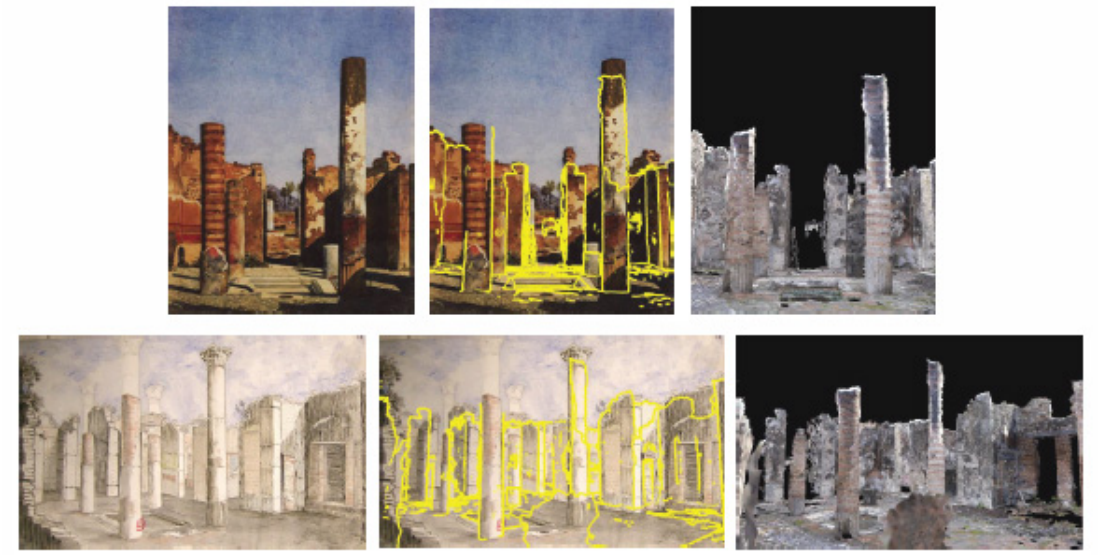


Figure 2-15: *Examples of alignment between the paintings and 3D model. For each example, left: painting; middle: 3D model contours projected onto painting; right: synthesized viewpoint from 3D model.[116]*

over time, the non-photorealistic nature of architectural paintings, and differences in the viewpoints used by the painters and photographers. This work combines the gist descriptor with the view-synthesis/retrival to obtain a coarse alignment of the painting to the 3D models. Figure 2-15 shows some alignment results between historical architectural paintings and 3D models.

2.4.2 General Solutions

Of the general solutions, Shechtman and Irani [121] propose to describe an image in terms of local self-similarity descriptors (SSD) that are invariant across visual domains. It is not to use the image feature appearance directly but instead to generate a correlation surface of local self-similarities from intensity patterns across the image. By comparing small patches extracted at each point in the image to their immediate neigh-

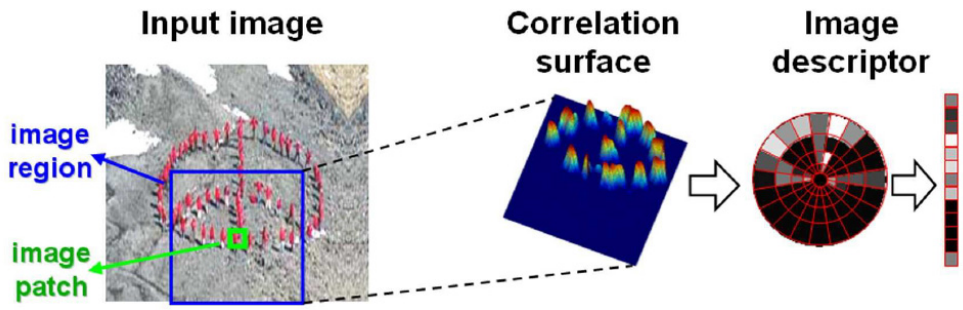


Figure 2-16: *Extracting the local self-similarity descriptor [121]*

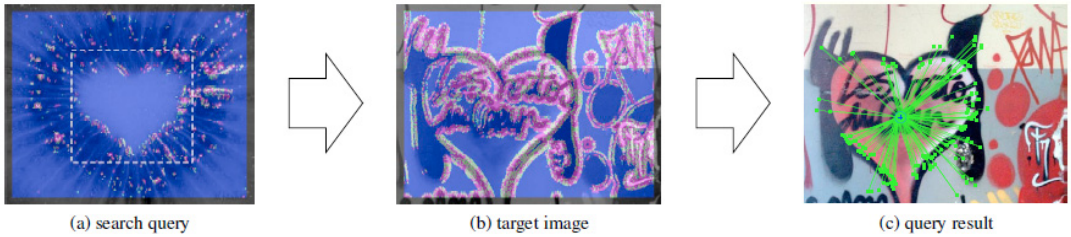


Figure 2-17: *An example of deformable shape retrieval using SSD [24]*

bourhood, the potential spatial description of image features which define the common shape can be extracted. This self-similarity representation gives a kind of abstraction when colour, texture and edges do not share between different objects, but share the same pattern. Figure 2-16 shows an example of local SSD extracted from an image.

Chatfield and Zisserman [24] extend SSD [121] to enable matching despite changes in scale. Further the descriptors are quantised to form a visual vocabulary, enabling the use of a bags of words approach for image retrieval. Figure 2-17 shows an example of shape retrieval using this method.

To break the restriction that SSD only can be applied locally, Deselaers and Ferrari [38] argue that self-similarity can and should be used globally rather than locally to capture long range similarities and their spatial arrangements. Figure 2-18 shows two selected patches and their global self-similarity (GSS), as the patch correlation with the entire image. Contiguous (patch 1) and repeating (patch 2) structures can be well recognized. Patch 2 shows that GSS can capture long range similarities within an image. The indirection characteristic of self-similarity results in similar patterns in the GSS images, although the original images appear very different. One drawback of GSS is that it is very expensive to compute if done directly, as every pixel in the image is correlated with the entire image.

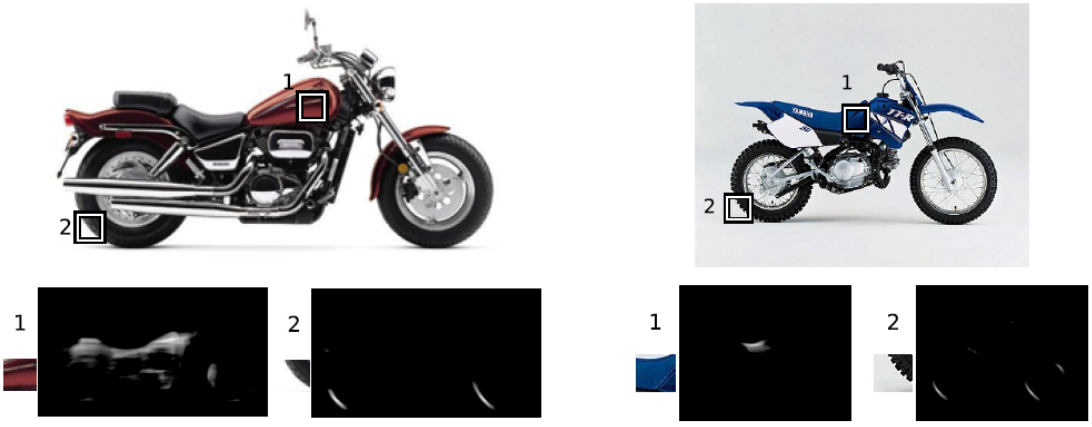


Figure 2-18: *Global self-similarity: self-similarity of two image patches with their respective images. [38]*

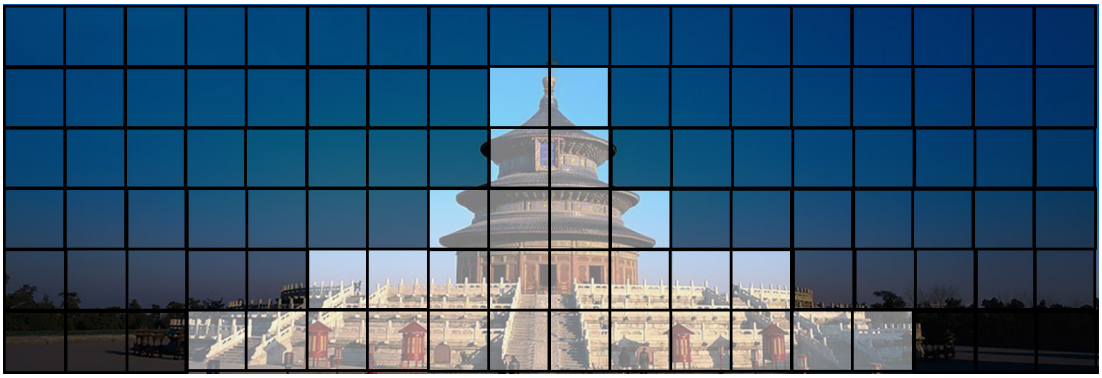


Figure 2-19: *The building is unique (covered by white square). The sky is quite common (covered by black square).*

According to Shrivastava et al [124], they do not propose any new descriptor but the main problem is: how to explore visual correspondence between two images that would be more in accordance with 'human expectations'. The solution is to design a visual similarity function that can determine which parts of representations will be used for matching. They hypothesize that the important parts of image are those that are more unique or rare within the visual world. For example, according to figure 2-19 , the sky in the image is not that unique as they are quite common in the natural world. It occurs everywhere; However, the building (Temple of Heaven) is more informative as it is more unique. Hence it is the building that will be the specific feature which can distinguish this image from the millions of natural world images. This is what the idea of 'data-driven uniqueness' is.

Specifically, they treat the query image as the 'positive sample', and set the rest of the images from natural world as the 'negative sample'. The central problem of this approach is to find the important feature of the 'positive sample' and use this feature to match with similar images in the retrieval set. Hence, this problem becomes to a discriminative learning problem. By using some gradient based feature descriptor (such as HoG), the query image can be represented as a vector of feature, and then the discriminative learning (such as SVM) produces a set of weights on these features of the images. Finally, the visual similarity between an input image(I_q) and other natural world images (I_m) can be defined as :

$$S(I_q, I_m) = W_q^T X_m, \tag{2.2}$$

In this equation, W_q is the query-dependent weight vector, X_m is the vector of the nature images.

Impressive results have been obtained for matching similar images of different depictions based on this study. Figure 2-20 shows some examples. However, there are some failure cases. For example, if the query scene is so cluttered that it is difficult for

Figure 2-20: *Examples of searching results by using Data-driven uniqueness techniques [124].*

this algorithm to decide which parts of the scene it should focus on. Moreover, speed remains the central limitation of this proposed approach, because it requires training an SVM with millions of negative examples at query time.

Auby and Russell [5] extend the idea of discriminative visual elements to a 3d scene. In this work they seek to automatically align historical photographs and non-photographic renderings, such as paintings and line drawings, to a 3D model of an architectural site. This work defines a discriminative visual element to be a mid-level patch that is rendered with respect to a given viewpoint from a 3D model with the following properties: (i) it is visually discriminative with respect to the rest of the "visual world" represented here by a generic set of randomly sampled patches, (ii) it is distinctive with respect to other patches in nearby views, and (iii) it can be reliably matched across nearby viewpoints. This definition is very close to the 'data-driven uniqueness' proposed in [124].

Most recently, Crowley and Zisserman [31] propose a framework to retrieval objects in paintings using discriminative regions. Similar to [124], they do not look the inside properties of paintings. Instead, they extend the work of [5] - the mid-level discriminative patches (MLDPS) and apply a RANSAC style algorithm to select a subsect of

Figure 2-21: *Example class images from the Paintings Dataset. From top to bottom row: dog, horse, train. [31].*

putative correspondences between the learned classifiers from MLDPS and objective painting regions and to make they are spatially consistent. Finally, [31] also investigates hybrid re-ranking strategies by using DPM detector [46]. With the help of DPM re-ranking, they show some interesting retrieval results on a 'Paintings Dataset'. However, we find the depiction styles in this dataset are quite single - only oil paintings are included. Figure 2-21 shows example classes from the dataset.

In [20], an experiment paper, Carneiro et al present a new databse of monochromatic artistic images containing 988 images with a global semantic annotation, a local compositional annotation, and a pose annotation of human subjects and animal types and provide an evaluation of several algorithms including BoF(Bags of Features), Structural Learning, Label Propagation, Inverted Label Propagation, Matrix Completion etc for image annotation and retrieval. Their experimental results show the Inverted Label Propagation performs better in global annotation.

Xiao et al [166] focused on using structures to represent objects depicted in different styles. They depend on spectral graph analysis of a hierarchical description of an image to construct a feature vector of fixed dimension. Hence structure is transformed to a feature vector, which can be classified using standard method such as GMM and SVM. Figure 2-22 shows the structure of objects extracted using graph energy analysis in [166].

Balikai and Hall [7] show great interests in depiction invariant image matching. In this work, a state-of-art segmentation is employed to segment the image to a hierarchy of regions, which is then used to build a hierarchical graph ; nodes of the graph are represented by the self-similarity descriptor (SSD). Then, an approach based on

Figure 2-22: *Examples objects and their parts, found using graph energy analysis. Facial parts are identified; four legged animals have their head and legs separated from their body; flowers have their centers extracted. [166].*

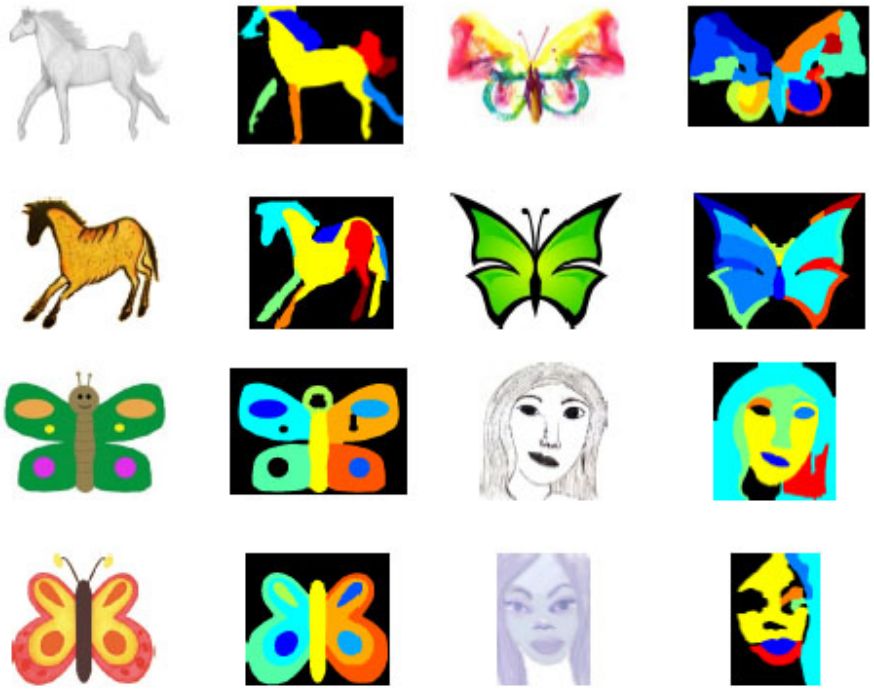


Figure 2-23: *Some examples of matched regions in image pairs depicted in different styles. Each pair of images is colour-coded to show regions that have been matched [7].*

maximising the overall quality of a Markov Random Field created by window search is introduced to match the graph. Following figure 2-23 shows some matching results by using this method.

Ginosar and Malik et al research on detecting people in Cubist Art. In [59], they evaluate existing object detection methods on some abstract renditions of objects, comparing human annotators to state-of-the-art object detectors on a corpus of Picasso paintings. Their results demonstrate that while human perception significantly outperforms current methods, human perception and part-based models exhibit a similarly graceful degradation in object detection performance as the objects become increasingly abstract and fragmented, corroborating the theory of part-based object representation in the brain. This conclusion keeps in line with the analysis we proposed in Chapter 5. Figure 2-24 shows some detection results for each method the authors compared in [59].

Deep features also shows good performance in addressing cross-depiction problems

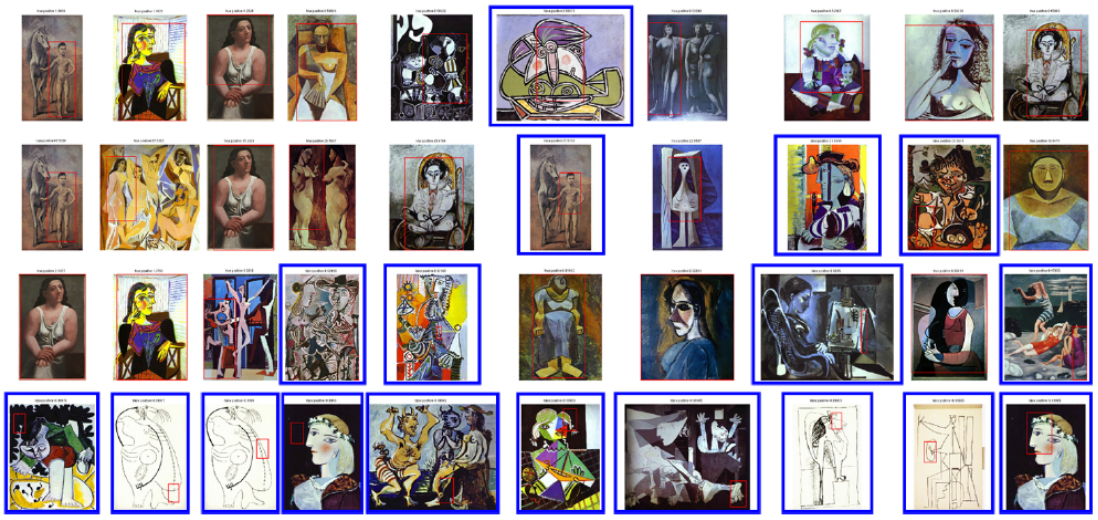


Figure 2-24: *Top ten detections for each method according to confidence from left to right. First row: DPM. Second row: Poselets. Third row: R-CNN. Fourth row: D&T (HOG). False positives are marked in blue. [59].*

recently. In [32], Crowley and Zisserman show that object classifiers, learnt using Convolutional Neural Networks (CNNs) features computed from virous natural images sources, can retrieve paintings containing these objects with great success. Specifically, a CNN network, which consists of 5 convolutional layers and 3 fully-connected layers, is trained solely using ILSVRC-2012 (Large Scale Visual Recognition Challenge). A feature vector of an image is obtained by passing it through the network and then the output of the penultimate layer is recorded. Then, linear-SVM classifiers are learnt using linear-SVM training data per class in a one-vs-the-rest manner.

2.5 Bridging the Literature Gap

Above sections first take a look at a few state-of-art methods in modelling visual object classes, showing that although these methods perform excellent in photometric domain, they do not consider the problem of recognising objects regardless of their depictions and most of them suffer the challenges of wide variations.

Inspired by the way of people drawing (using simple shapes), we then investigate the literature studying planar shapes in computer vision in 2.2. Although shape has been put to use in many computer vision tasks not limited to matching, classification and retrieval, none of the literature we know asks as we do: " is there a set of shapes commonly present in natural images?". And we propose to use these simple common shapes in cross-depiction objects modelling task.

For structure, human are able to classify objects because one can recognise object structure in the brain. This finding has lead to the fashion in computer vision that

of using a structural representation to model a visual object class. We reviewed some representative works of this scope in the section 2.3, including DPM [46], a robust part-based structural representation method. However, there are still limitations of using DPM to represent object across different depictive styles, for example, the wide variation of local features in different depictions is hard to be learned in a single model. We propose to use a multi-label system to fill this gap.

Section 2.4 reviewed some studies on cross-depiction problems. Among these works, some of them are only focused on one specific domain, such as [53, 72, 90] and some designed local only descriptors [121, 24]. Some other works like [124, 5, 31] are learning discriminative regions of an image instead of analysing properties of paintings.

In this thesis, we take the inspiration from human vision, painting methods and psychology to investigate the common properties in both photos and art. Based on the investigation, we propose a number of methods to model visual object classes across different depictive styles and use the model to detect and classify objects in a challenging photo-art dataset built by ourselves. We make our efforts to bridge the gap between photos and art works in several tasks of Computer Vision.

CHAPTER 3

PRIMITIVE SHAPES

3.1 Introduction

Shape has been well studied in many disciplines, yet to the best of our knowledge the question as to whether there is a set of elementary planar shapes that appear commonly in the world around us has never been asked. In this chapter, we describe an experiment designed to test the following hypothesis: *some regions in image segmentations can be fitted as one of a few primitive shapes.* Not wishing to force regions into classes, we developed a classifier (with parameters such as bandwidth of clustering method) designed to find clusters of a size greater than would be expected if the shape of regions were randomly generated. We used two different "shape spaces" (*i.e.* shape descriptors), three different segmentation methods, and three image databases. It concludes that the most common of those found are familiar enough to be named: shapes such as triangles, squares and circles (more exactly, these shapes up to affine transformation). We propose to use qualitative shapes as features in future applications.

In Chapter 1, we stated that psychologists have explicitly used shape as a primitive to explain cognition in the form of Geons, a concept which comes from Biederman's theory [13]. Geons are the simple 2D or 3D forms such as cylinders, bricks, wedges, cones, circles and rectangles corresponding to the simple parts of an object of object recognition. (These correspond almost exactly to art practice, especially that of Leger.) The theory proposes that the visual input is matched against structural representations of objects in the brain. These structural representations consist of Geons and their relations (e.g., an ice cream cone could be broken down into a sphere located above a cone).

In the early of 20th century, some schools of art and individual artists keen to use simple regular shapes (in particular rectangles, ellipses and triangles) as basic constructs for painting. The artists found theses primitive shapes are sufficient to

describe the world producing abstract and figurative artworks. This not only happens among artists, even people without any formal training of drawing can use simple shapes to layout scenes or objects. For example, a simple representation of a face can be constructed using circles alone and this idea can be extended to almost any objects - bicycles, chairs, houses can all be represented using primitive shapes. This proves that these shapes make a powerful but simple descriptive set.

Moreover, empirical evidence that aligns with artistic intuition has existed since at least the 1970s, when psychologists such as Rosch [113] showed simple shapes (specifically triangles, squares, and circles) are easier for humans to recall other shapes. It is interesting to speculate that this may explain why humans have words to describe these shapes, and it is also interesting that our experiments show it is exactly these shapes that occur in natural images with a frequency which is well above that expected by chance alone.

Some others also relished the important role the simple shapes played in bridging the gap between photos and paintings, for example, Balikai et al [8] fit a shape drawn from a selection of shape families to regions that segment an image, using a supervised classifier. And then they use results from the classifier to match photographs and artwork of particular objects using a few qualitative shapes. Figure 3-1 displays some matching results from [8].

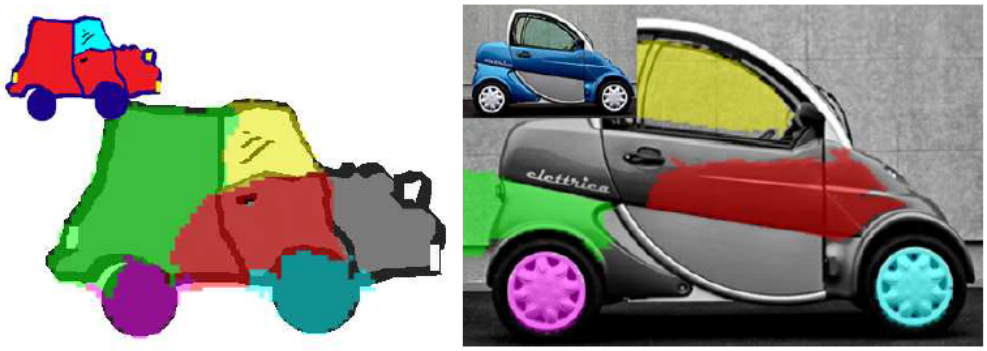


Figure 3-1: *Parts of a drawn car and parts of a photographic of a car are matched in [8], as shown by the colour coded regions.*

Song et al [130] propose to use fitted qualitative shape labels for the purpose of generating synthetic abstract art from photographs. Figure 3-2 shows how a photograph is rendered into a piece of artwork where paper cutouts were used as basic elements.

Contributions

Work presented in this chapter has been published in BMVC 2012 [163].

- Providing empirical evidence that some regions in segmented images can be classified or fitted as one of a few primitive shapes, upon given appropriate region

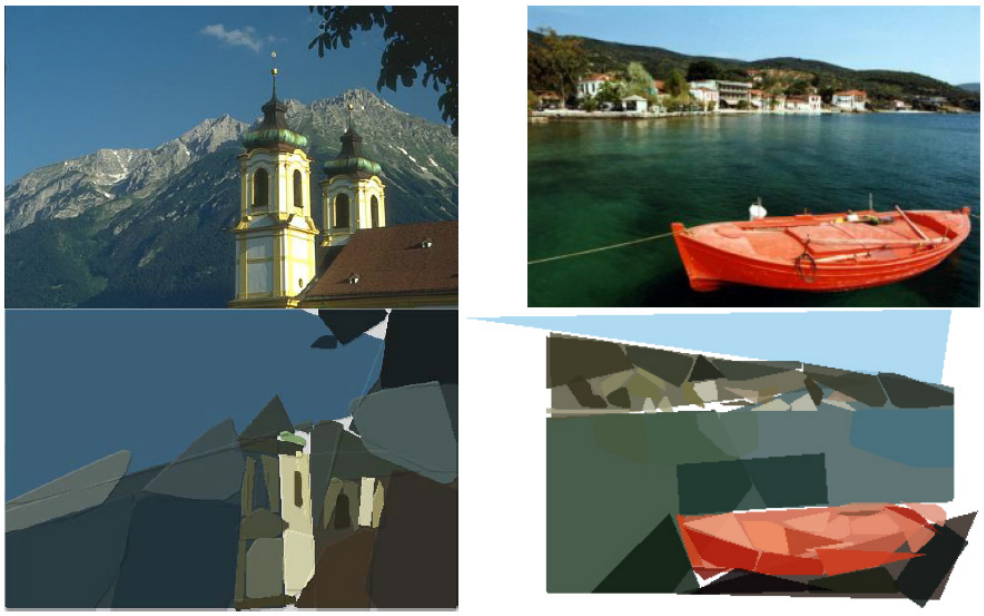


Figure 3-2: *Examples of photo to art transfer using simple shapes to fit segmented regions [130]. Top: original photos. Bottom: paintings rendered as paper cutouts.*

descriptions and well-designed classifiers.

- A classifier to fit primitive shapes to segmented regions of an object.
- A computationally efficient classifier to categorise scenes based on ration of primitive shapes.
- A new agglomerative clustering method to cluster similar binary shapes.

In the following, we represent our experimental framework and method with experimental results in section 3.2 and section 3.3, and the additional experiment in appendix A. Section 3.4 reports an application, scene classification, based on our research and the conclusion of this chapter is presented in section 3.6.

3.2 Experimental Method

Our experiment is designed to test the hypothesis that *some of regions in image segmentations can be classified as one of a few primitive shapes.* Supervised methods for training a classifier are ruled out, because our aim is to discover classes should they exist rather than to force regions into classes. So we used an unsupervised classifier. To guard against bias in shape description we use two different descriptors, Zernike moments and Chebychev moments and experimented extensively with one of them to show the choices we made have little or no impact (see Appendix A).

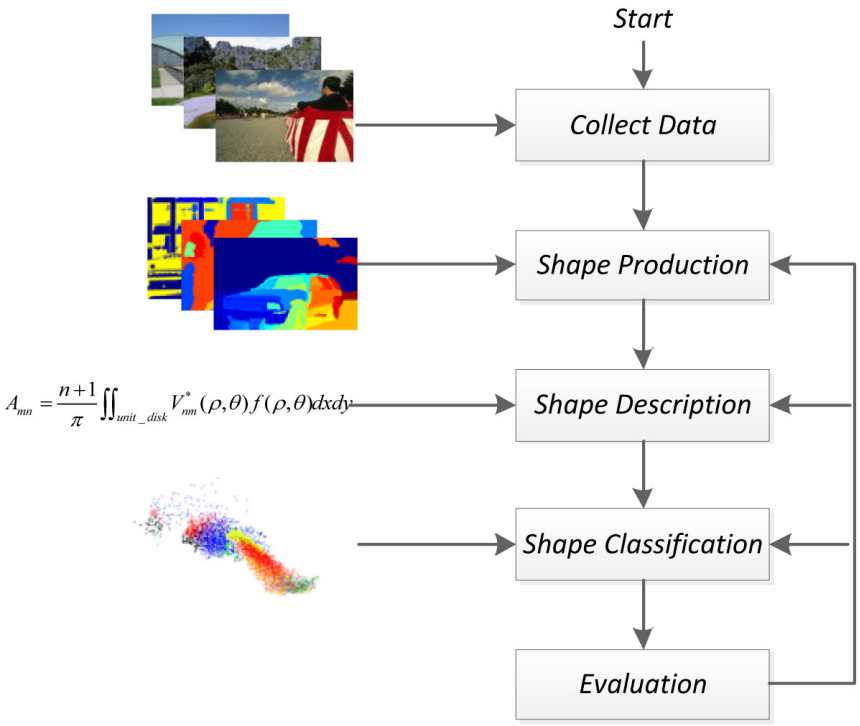


Figure 3-3: *Experimental Framework*

We used a range of segmentation methods, none of which force any shape on regions. We used three different databases, two publicly available, the other specifically designed by us for our experiment. Our approach is to automatically cluster (with an data-driven bandwidth selection scheme) regions that have been segmented from images, and compare these clusters with those created from a database of randomly created regions. Figure 3-3 illustrates our method.

No matter how we configured our experiment, we concluded that shapes classes we can call triangles, squares, and circles do exist in natural images. Interestingly, the ratio of these shapes in any given picture can be used to predict the class of the scene as whole, that is we have an immediate application in scene categorisation, as explained in Section 3.4.

3.2.1 Three Image Databases, and a Random Generator

Our experiments are based on three images databases, chosen because they offer a diverse set of content. We also used a database of shapes created at random. Typical images from these databases can be seen in Figure 3-4. The first two databases come from MIT and Berkeley; the third has almost no human made objects – we call this database 'Bath Nature'.

The ***MIT database*** is publicly available [80]. Designed for eye tracking experiments, the database contains 1003 images, including street scenes, buildings, animals

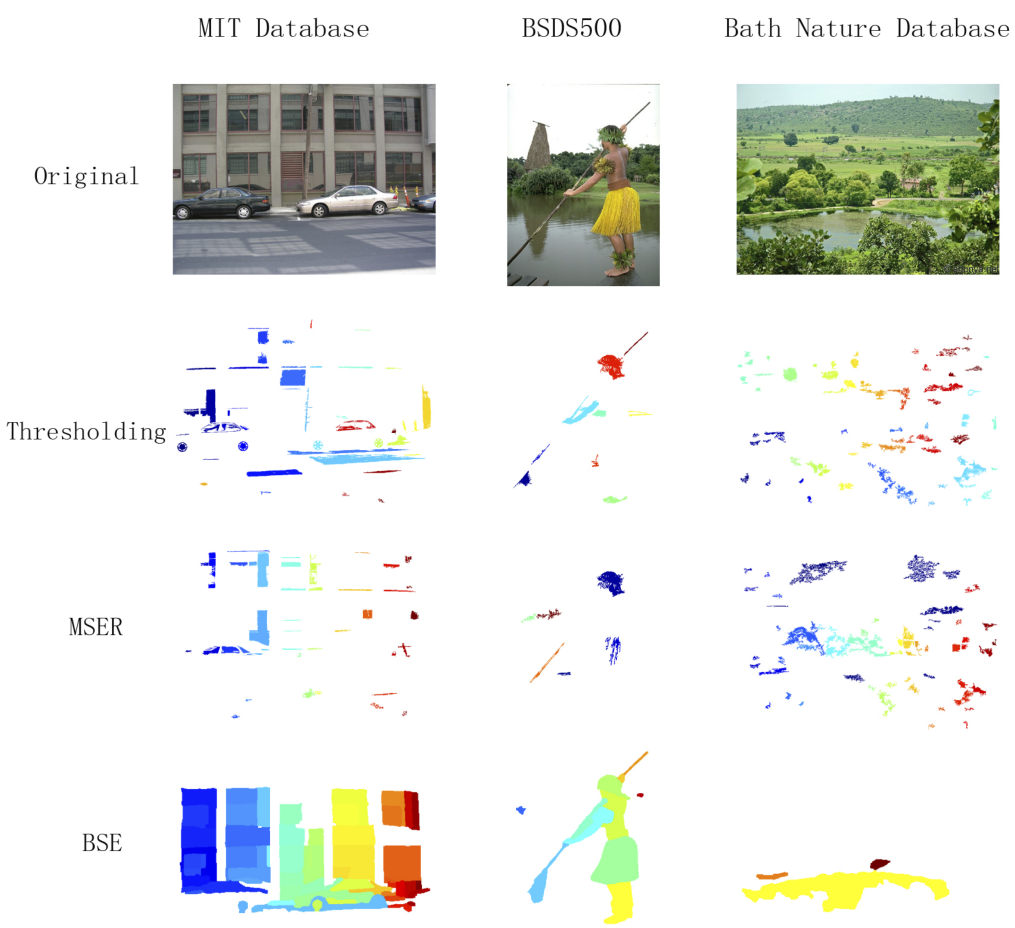


Figure 3-4: *Regions produced by different segmentation algorithms over different image databases.*

and natural landscapes *etc.* We randomly chose 200 images as training data, and 100 for testing data.

The ***Berkeley Segmentation Database (BSDS500)*** [96] is a well known, publicly available database often used for experiments in contour detection and image segmentation. It includes 500 pictures, most of them are natural images, but also includes faces and animals. We randomly choose 200 images as training data, and 100 for testing data.

The ***Bath Nature Database*** has 50 pictures of outdoors; forest, field, seascapes *etc.* There are very few human-made objects. Human made objects such as buildings, cars, and indeed just about all other manufactured objects are often constructed using primitive shapes: wheels, bricks, windows, *etc.* Our database tries to avoid such objects to eliminate such bias.

In addition to fixed databases we generate ***Random Regions*** so that we have a baseline for the size of classes formed by clustering random regions. The purpose of

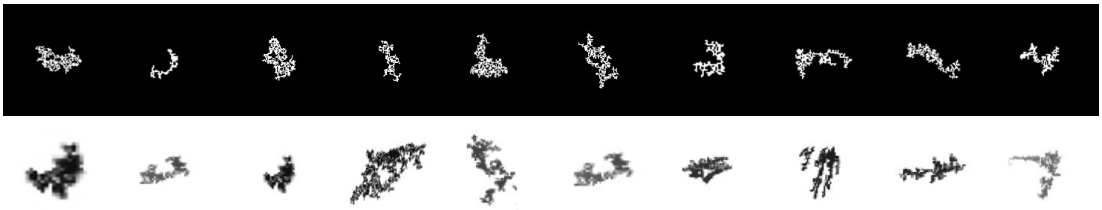


Figure 3-5: *Typical random shapes. Upper-side row shows the regions produced by our random shape generater. Lower-side row shows regions harvested from different segmentors.*

generating random regions is to help to identify the primitive shape classes. To create random regions we developed the following algorithm:

(i) Create an $N \times N$ image of independent random numbers.

(ii) Choose the central pixel to be the current region centre.

(iii) Mask all pixels that form the outer border of the current 4-connected region, and add the masked pixel with the highest random number to the region.

(iv) Continue until the region has the required number of pixels.

The halting number is randomly drawn from a uniform distribution over $[100, 600]$. The lower bound is used because we filter out regions with less than 100 pixels, the upper bound represents the size of a typical middle sized region in segmented image regions.

Figure 3-5 shows the regions generated at random by using the above algorithm, and regions classified as random shapes from image segmentations by using a shape classifier implemented in Section 3.4.1. It suggests that random regions produced by our generator are close to those in the real image datasets.

3.2.2 Three Segmentation Algorithms

To offset bias regarding any particular segmentation algorithm we used three; one very simple, one popular, one state of art. In each case any region that touched a picture boundary or which contained less than 100 pixels was removed from further consideration. The first was to remove any bias introduced by straight boundary edges, the second to remove noise — needed primarily for segmentation by thresholding. Typical segmentation output can be seen in Fig 3-4. We harvested about 10^4 regions from each database, for each segmentation algorithm.

Thresholding is perhaps the simplest methods for image segmentation. A grayscale image, I maps to a binary image: $b = I > \tau$, for threshold τ. Assuming gray values in $[0, 1]$, we set $\tau = 1/2$. We used both black and white regions.

Maximally Stable Extremal Regions (MSER) are regions found by analysis of successive threshold images [97]. The MSER algorithm extracts from an image I a

number of co-variant regions, called MSERs. An MSER is a stable connected region of some level sets of the image I. The formal definition of MSER can be found in [97]. In the original formulation, MSERs are controlled by a parameter Δ, which controls how the stability is calculated. With the increase of Δ, fewer stable regions are detected. We set $\Delta = 1$ in our experiment. Other parameters are settled following [147], maximum variation is 0.25, minimum diversity of region 0.2. And both dark-on-bright regions and bright-on-dark regions are detected.

The ***Berkeley Segmentation Engine(BSE)*** is based on the probability of boundary (Pb) maps introduced by Arbelaez *et al* [4]. It is considered as one of the most successful segmentation techniques because it compares very well against human produced ground truth using the Berkeley Segmentation Dataset (BSDS-500). Probability boundary predicts the posterior probability of a boundary with orientation at each image pixel by measuring the difference in local image brightness, color, and texture channels. In order to detect fine as well as coarse structures, they consider gradients at three scales, then they linearly combine these local cues into a single multi-scale oriented signal and maximize over orientations yields a measure of boundary strength at each pixel. Finally, with a simple linear combination of the most salient curves signal, global probability boundary is generated. We follow [4] to use the default parameters setting for BSDS500, which has produced best boundary estimation results against the ground truth.

Before we go to the next step, we want to first verify our artificial generated random shapes share the similar distribution with those regions segmented from natural images by using above three segmentations, we examine the spectrum for each image database that we used and our random shape database. This is valuable since if the spectra can not be matched, it means the dataset of random shapes is a different distribution in the statistical sense with the real image datasets. Then, comparing the natural images with random shape to find the significant shapes will be meaningless. We design the further experiment to test this.

Matching the Spectrum

We use the Fourier transform to transfer the shapes to spectra at first. Let s_i be a binary shape mapped to the unit disc and let $f(s_i)$ be its 2D fourier transform. Then the average absolute spectrum is

$$f_a = \frac{1}{N} \sum_{1}^{N} |f(s_i)| \tag{3.1}$$

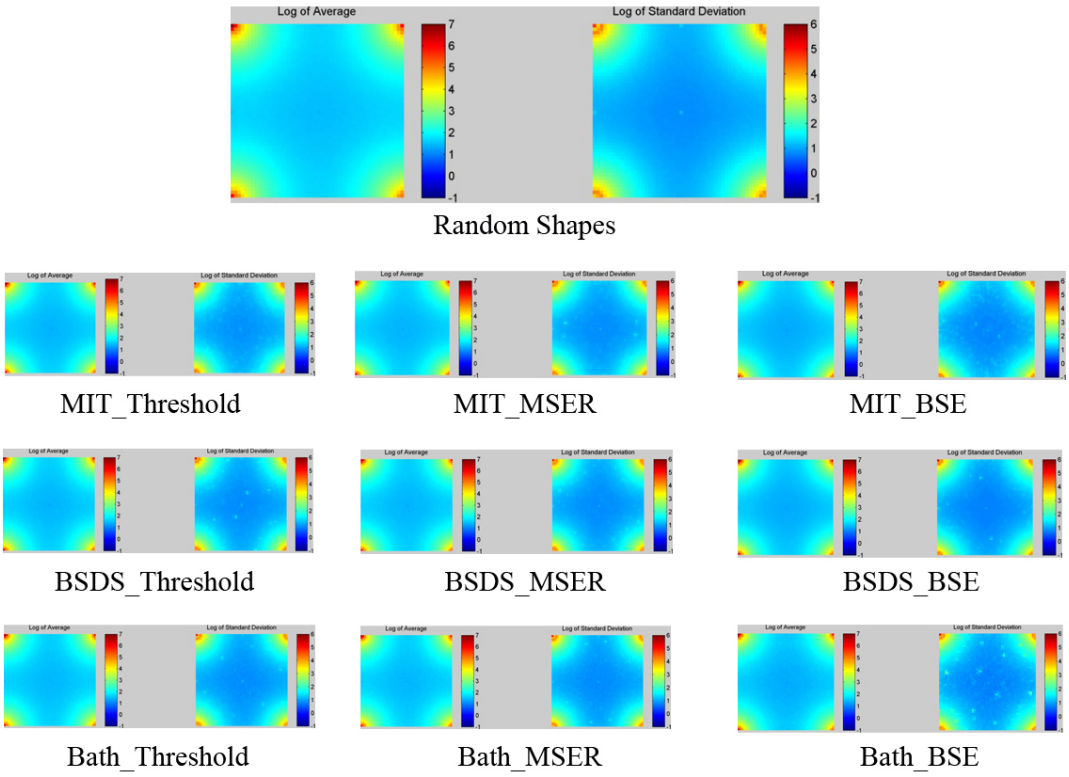


Figure 3-6: *The log of average and standard deviation of 2D fourier transform of random dataset and other database. The RMS distance from the random spectrum to the database spectrums. Table 3.1*

and the standard deviation is

$$f_s = \sqrt{\frac{1}{N}\sum_{1}^{N}(f(s_i) - f_a)^2} \tag{3.2}$$

We compute the f_a and f_s for each database and different segmentation methods and random shape database. Then, we display the log of the result for each dataset, showing in figure 3-6. we can find that all the datasets, including random shape datasets, have very approximate average and standard deviation. This suggests the spectrum for each image database and each segmentation method and random shapes may be matched. We also quantified this similarity by using the Chernoff distance.

In statistics, the Chernoff distance measures the similarity of two discrete or continuous probability distributions. It reflects the probability that two distributions are the same. If the Chernoff distance is 1, then the two distributions are the same. And while the distance is 0, there will be nowhere intersect between two distributions.

The table 3.1 shows the Chernoff distance between segmented shapes by using different segmentation methods. A value of 0.8 or so means that about 80% of the time we

	BSE	MSER	Threshold	Random
BSE	1	0.8366	0.8562	0.5013
MSER	0.8366	1	0.9807	0.8307
Threshold	0.8562	0.9807	1	0.8219
Random	0.5013	0.8307	0.8219	1

Table 3.1: The Chernoff distance between each dataset.

will confuse the spectrum of random shapes for the spectrum of segmented shapes. And this means the spectrum of these datasets are very similar. And an interesting observation is that the distribution of shapes which are generated by Berkeley segmentation does not have as high similarity as other distributions. This suggests that Berkeley segmentation is good at generating more regular shapes than other segmentors (see figure 3-4).

3.2.3 A Whitening (Affine) Transform and Re-sampling

We normalise each region (shape) before computing its description. We apply a whitening transform that brings the region into the unit disc, as follows. Let $X = \{x_i\}$ be points of a region, with $\bar{x}$ their centroid and $C = ULU^T$ their covariance. Then

$$y_i \quad = \quad L^{-1/2}U^T(x_i - \bar{x}) \tag{3.3}$$

is a whitening transform, which applies an affine transform to the shape by centering it at the origin, rotating it to a canonical frame and differential scaling over each eigenaxis. This will map any triangle into equilateral form, any rectangle into a square, and any ellipse into a circle.

The new shape will have a unit covariance in each eigendirection, so we scale by the point most distant from the origin to map the shape into the unit disc. Scaling into the unit disc changes the effective sample rate. To make sure that this plays no role in moment computations, we re-sample the shapes into a 50^2 regular grid. To make a binary image of the original shape we consider each point x_i in the original to be the centre of a radially symmetric Gaussian of width 1. This maps to an elliptical Gaussian with eigenaxis U and covariance L/s^2. An anti-aliased version of the transformed region is now given by

$$f(y) = \sum_{i=1}^{N} \exp\left(-\frac{1}{2s^2}(y - y_i)^T ULU^T (y - y_i)\right). \tag{3.4}$$

where $s = \max_j |y_j|_2$ and $y_i \leftarrow \frac{y_i}{s}$. We threshold this to obtain a binary region and $f(y_i) = 1$ if $f(y_i) > \bar{f}$. In which $\bar{f}$ is the average value of $f(y)$. It is these binary regions that we describe with a standard descriptor, then classify.

3.2.4 Two Shape /Region Descriptors

There are many shape descriptors to choose from; we opted for Zernike moments [82] and Chebyshev moments [109]. These moments operate over sets of points, so are useful for describing solid regions of the kind produced by typical image segmentation algorithms. They are fast to compute, again useful when faced with many regions in an image segmentation. The forms of these moments we use are invariant to rotation and robust to noise.

Zernike moments

Zernike moments [82] are constructed by using a set of complex polynomials which form a complete orthogonal basis set defined on the unit disk. They are parameterised by two integers; $n \geq 0$, and m such that $|m| \leq n$ and $n - |m|$ is even. In polar coordinates, (ρ, θ), the $(n, m)^{th}$ Zernike basis function, $V_{nm}(\rho, \theta)$, defined over the unit disk is

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta), \ \rho \leq 1, \tag{3.5}$$

in which $j = \sqrt{-1}$. The Zernike radial polynomials, $R_{nm}(\rho)$, are defined as:

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} -1^s \frac{(n-s)!}{s!(\frac{(n+|m|)}{2} - s)!(\frac{(n-|m|)}{2} - s)!} \rho^{n-2s} \tag{3.6}$$

For a binary image $f(x, y)$, the mn^{th} Zernike moment is

$$Z_{mn} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V^*_{nm}(\rho, \theta), \tag{3.7}$$

with $\rho = \ x^2 + y^2 \leq 1$. We use all moments up to $n = 6$. (This choice is justified in Appendix A), and $m \in [-n, n]$, giving $(n + 1)(n + 2)/2$ basis functions.

We normalise the Zernike moments as follows: (i) we use the absolute value, so that the moments are invariant to rotation; (ii) we divide by the zeroth order moment, so that the moments are invariant to pixel area.

Chebyshev moments

Chebyshev moments [109] depend on Chebyshev radial polynomials of the second kind are defined as:

$$R_n(\rho) = \sqrt{\frac{8}{\pi}} (\frac{1-r}{\rho})^{1/4} \sum_{s=0}^{n/2} (-1)^s \frac{(n-s)!}{s!(n-2s)!} (2(2\rho - 1))^{n-2s} \tag{3.8}$$

where n is a non-negative integer. Then, the Chebyshev moment is defined by:

$$C_{mn} = \int_{unit\ disk} R_n(\rho) f(\rho, \theta)\ d\rho d\theta \tag{3.9}$$

in which $f(\rho, \theta)$ is a binary image in radial polar coordinates.

3.2.5 Clustering

The problem now is to find clusters in a collection of shapes, in a *fully unsupervised* way. We proposed an agglomerative clustering based on shape correlation. However, it is intractable for agglomerative to cluster 10^4 shapes as it depends on pair-wise interactions. Hence, a coarse clustering has to be done to reduce the number of pairs. We choose mean shift clustering, which is fast, well known, and is non-parametric. We locate mean shift clusters that are statistically significant, typically about 30 to 40 clusters, each with an associated canonical shape. It reduces the number of pairs from about $(10^4)^2$ to a more manageable 35^2, approximately. The procedure of clustering is outlined below (Seeing Algorithm 1).

Mean Shift Clustering

Mean Shift is a tool for finding modes in a set of data samples, manifesting an underlying probability density function (PDF). It is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters.

Given n data points $x_i, i = 1, ..., n$ on a d-dimensional space R^d, the multivariate kernel density estimate obtained with kernel $k(x)$ and window radius h is

$$f(x) = \frac{1}{nh^d} \sum_{1}^{n} k(\frac{x - x_i}{h}) \tag{3.10}$$

For radially symmetric kernels, it suffices to define the profile of the kernel $k(x)$ satisfying

$$K(x) = c_{k,d} k(||x||^2) \tag{3.11}$$

where $c_{k,d}$ is a normalization constant which assures $K(x)$ integrates to 1. The modes of the density function are located at the zeros of the gradient function $\nabla f(x) = 0$. The gradient of the density estimator is

$$\begin{aligned} \nabla f(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x) g(||\frac{x - x_i}{h}||^2) \\ &= \frac{2c_{k,d}}{nh^{d+2}} [\sum_{i=1}^{n} g(||\frac{x - x_i}{h}||^2)][\frac{\sum_{i=1}^{n} x_i g(||\frac{x-x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x-x_i}{h}||^2)} - x] \end{aligned}$$

where $g(s) = -k'(s)$. The first term is proportional to the density estimate at x computed with kernel $G(x) = c_{g,d}g(||x||^2)$ and the second term

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g(||\frac{x-x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x-x_i}{h}||^2)} - x \tag{3.12}$$

is the *mean shift*. The mean shift vector always points toward the direction of the maximum increase in the density. The mean shift procedure, obtained by successive computation of the mean shift vector $m_h(x^t)$ and translation of the window $x^{t+1} = x^t + m_h(x^t)$, is guaranteed to converge to a point where the gradient of density function is zero.

To use mean-shift we project all of the descriptors into a deflated eigen-space (aka. principal component analysis) to reduce the original 27-dim descriptor to about 17 dimensions (keeping 97% eigenenergy). A whitening transform ensures the data exhibit a unit standard deviation in each eigen-direction: now a single number can control the bandwidth of a mean shift clustering algorithm, because the data are equally spaded in all directions. Even though mean-shift is a non-parametric algorithm, it does require the bandwidth parameter h to be tuned. However, other than setting a fixed bandwidth, we apply a data-driven scheme to decide the value. The bandwidth is automatically set to be

$$\rho = \frac{1}{3}\left(\frac{v}{N}\right)^{1/n}, \tag{3.13}$$

in which v is the hyper-volume of the bounding box enclosing the N data points, which exist in a n dimensional space. This is the characteristic radius of a hyper-sphere surrounding each datum, assuming they are uniformly distributed, but scaled because we observed that most of the points were clustered in about 1/3 of the hyper-volume, in each direction.

Mean shift yields many clusters of different sizes. Some clusters contain hundreds or even thousands of shapes, others contain just one or two. In order to decide which clusters are statistically significant we produce 10^4 random shapes, seeing Section 3.2.1. We clustered the random shapes using mean shift, and found that no cluster size exceeded about 10; the vast majority were singletons, seeing Figure 3-7. This suggests that the segmented regions clusters can be discriminated from generated random region clusters.

Given this result, to locate statistically significant clusters in shapes drawn from an image database we count the total number of shapes in a cluster of a given size to get $p(m|D)$, which is the probability of observing a cluster of size m, given source $D \in \{\text{Image Database}, \text{Random}\}$. We keep only those clusters of size m for which

$$p(m|\text{Image Database}) > p(m|\text{Random}). \tag{3.14}$$

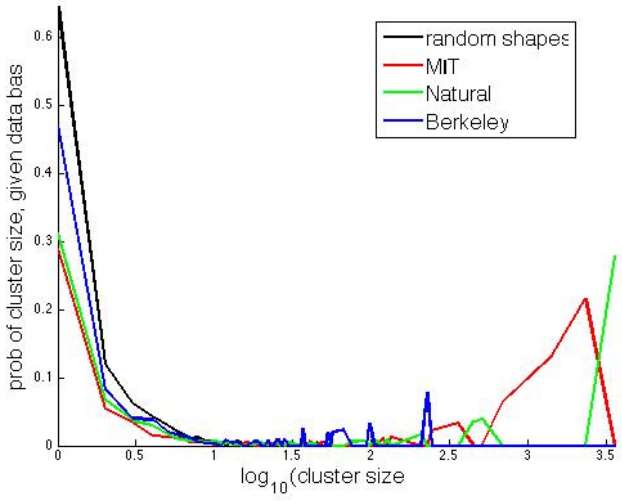


Figure 3-7: *The probability of number of shapes in a group of a given size for different databases using threshold segmentation and Zernike moments. Random shapes do not create clusters of more than about 10 shapes.*

There are typically around 30 to 40 such clusters, which together contain between 35% and 80% of all shapes, depending on the image database and segmentation method, seeing Table 3.2 and Table 3.3.

Given the additional experiments regarding choice of moment descriptors, in appendix A, this result is enough to have confidence that our hypothesis is true: simple shapes do exist to a statistically significant degree in real-world images, upon given appropriate region descriptions and well-designed classifiers.

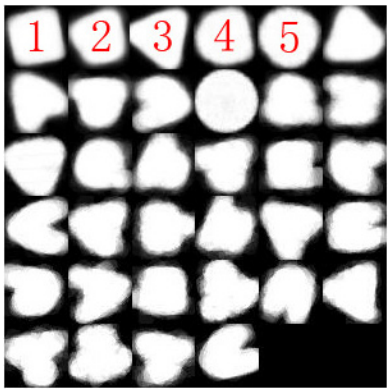
Agglomerative Clustering

Mean-shift yields a few tens of clustered shapes, that number can be reduced to less than ten by agglomerative clustering, using a method developed by ourselves. We begin by rotating all (whitened) shapes in a cluster to the first. Now $s(x, y, i, j)$ denotes a point in the i^{th} aligned shape in the j^{th} cluster. It is now easy to compute the mean shape, and to estimate the spatial error distribution:

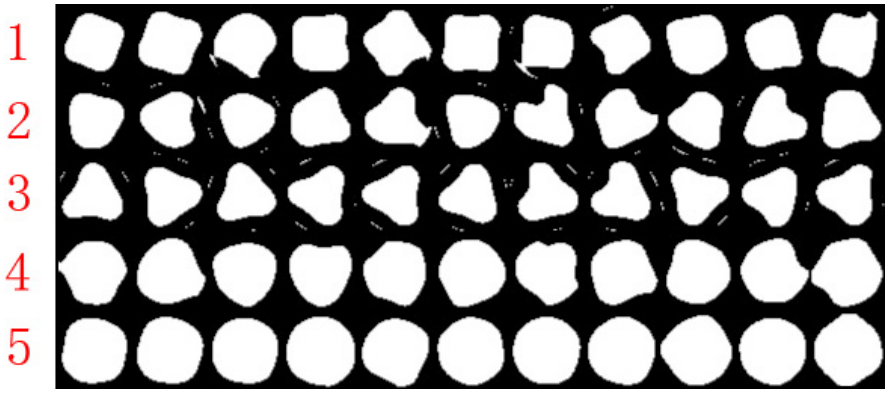
$$m(x, y, j) = \frac{1}{N_j} \sum_{i=1}^{N_j} s(x, y, i, j) \tag{3.15}$$

$$e(x, y, j) = \left(\frac{1}{N_j} \sum_{i=1}^{N_j} (s(x, y, i, j) - m(x, y, j))^2 \right)^{1/2} \tag{3.16}$$

where N_j is the number of shapes in class j. The error image, e, locates where the class varies most — which invariably is at the boundary. We normalise e so the image



(a)



(b)

Figure 3-8: (a) *Average shapes from mean shift clusters.* (b) *Some shape examples from different mean shift clusters, the column number corresponds to the marking number of clusters in* (a).

sums to unity. The thresholded mean shape

$$\mathsf{icon}(x, y, j) = m(x, y, j) \geq \text{mean}[m(x, y, j)] \tag{3.17}$$

acts as an *icon* for the binary shapes in the class. Typical mean shapes coming from mean shift can be seen in Figure 3-8 (a), which informally suggests regions of similar shape form clusters. We must now combine classes, so need a class descriptor. And Figure 3-8 (b) shows shape examples from different mean shift clusters.

Our class descriptor uses the boundaries pixels of the icons, calling these $b(x, y, j)$. We rotate a boundary image about its centre and at each angle, θ, computing its *similarity* to the error image in the same class using

$$\phi(\theta, j) = \sum_{xy} e(x, y, j)b(x, y, j, \theta) \tag{3.18}$$

$\phi(.,.)$ is now a one-dimensional signal that characterises an icons rotational symmetry

Algorithm 1: Clustering (Mean shift and Agglomerative)

Input : Image Database Z, Random Shape Data R
Output: Primitive Shape Classes $\mathscr{P}$

run *mean shift clustering* on Z and R
for $i := 1$ **to** *num_clusters_Z* **do**
  if $p(m|Z) > p(m|R)$ **then**
    Add cluster c_i to the Primitive Shape Classes $\mathscr{P}$
  end
end
repeat
  for $j := 1$ **to** $size(\mathscr{P})$ **do**
    calculate $\phi(\theta, j)$ (Eq.3.18)
  end
  calculate inter-class similarity matrix (Eq.3.19)
  merge process (Eq.3.20) and **update** $\mathscr{P}$
until *size($\mathscr{P}$) not change*;

against its own error set. Next, we compute the maximum of the normalised cross correlation between pairs of classes, to obtain an inter-class similarity score:

$$c(j,k) = \max_{\alpha} \frac{\sum_{\theta}(\phi(\theta,j) - \bar{\phi}(\theta,j))(\phi(\theta-\alpha,k) - \bar{\phi}(\theta,k))}{(\sum_{\theta}(\phi(\theta,j) - \bar{\phi}(\theta,j))^2)(\sum_{\theta}(\phi(\theta-\alpha,k) - \bar{\phi}(\theta,k))^2)} \tag{3.19}$$

This is not a symmetric function, so that $c(j,k) \neq c(k,j)$ in general. We set $c(i,i) = 0$. We merge classes j and k only if their inter-class scores are such that they share a mutually maximal class:

$$\left(\arg\max_i c(j,i) = \arg\max_i c(i,j)\right) = \left(\arg\max_i c(k,i) = \arg\max_i c(i,k)\right) \tag{3.20}$$

This ensures the pair of classes are tightly bound. In practice, we can group several clusters simultaneously because a single icon may be mutually maximal with several others, so that this form of agglomerative clustering is very efficient. All shape classes within a single group are bundled into one, aligned, and a new mean and error image is computed by weighted sums. For example

$$m(x,y,j') = \sum_j \frac{N_j m(x,y,j)}{\sum_j N_j}, \tag{3.21}$$

similarly for error images. Agglomerative clustering halts when there is no change in the number of shape classes.

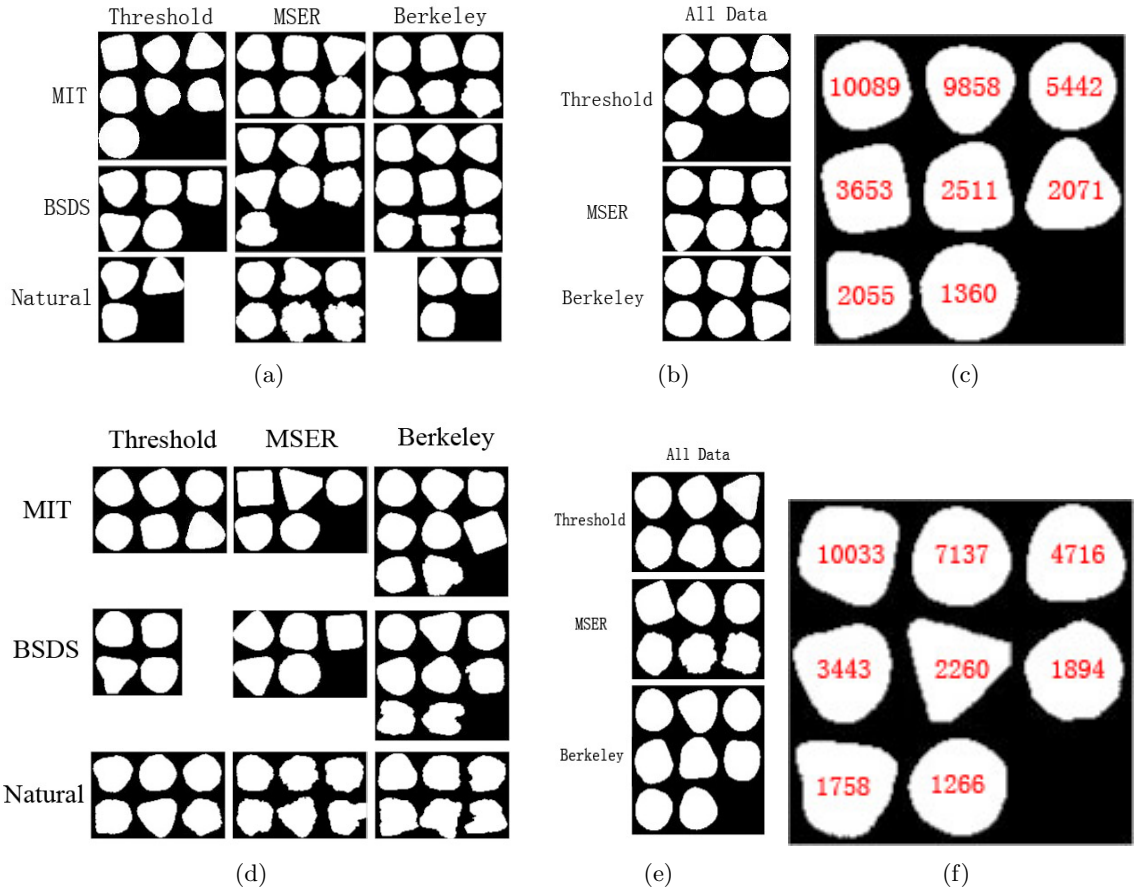


Figure 3-9: *Matrices of final results (*Upper*: Zernike Moments,* Lower*: Chebyshev Moments). In each matrix element the shapes are ordered by descending frequency from top-left to bottom-right. (a,d) Each entry shows the shape icons yielded by different databases, different segmentation methods. (b,e) Shape icons for different segmentation methods yielded by combining all three databases, different segmentation methods. (c,f) Final grouping result by combing all databases and segmentations. The number of each primitive shape is plotted in each corresponding icon.*

3.3 Experimental Results

Final shapes for each database and each segmentation method can be seen in Figure 3-9. The shapes tend to be simple — and nameable shapes such as circles, squares and triangles are common. In some cases we also see a square under a homography, which lies between square and triangle in feature space, and a simple shape lies between the square and circle, we conjecture it is a composite of higher order regular polygons. There are some irregular looking shapes too, but these are not often observed compared to the regular shapes. The fractional number of these 'primitive' shapes depends on segmentation and database, but is consistently high; as Table 3.2 and Table 3.3 shows, over 1/2 of all segmented regions fall into one of the discovered categories.

We also display the random shapes data and primitive shapes classes data in the

Zernike	Thresholding	MSER	Berkeley
MIT	66.65%	67.12%	81.45%
BSDS	59.92%	54.60%	80.65%
Natural	62.64%	36.24%	60.36%

Classified Shapes	37039
Un-Classified Shapes	19953
Classification Fraction	64.99%

Table 3.2: *Results obtained using Zernike moments.* Upper: *The percentage of 'primitive shapes' we detected as being statistically significant amongst total shapes detected from the MIT,BSDS500 and our Natural database, by using different segmentation algorithms.* Lower: *Number of classified and un-classified shapes from all three database, and the fraction of classified shapes with total shapes.*

Cheby	Thresholding	MSER	Berkeley
MIT	52.55%	57.57%	52.06%
BSDS	44.90%	49.89%	46.86%
Natural	49.93%	48.38%	64.30%

Classified Shapes	32507
Un-Classified Shapes	24485
Classification Fraction	57.04%

Table 3.3: *Results obtained using Chebyshev moments.* Upper: *The percentage of 'primitive shapes' we detected as being statistically significant amongst total shapes detected from the MIT,BSDS500 and our Natural database, by using different segmentation algorithms.* Lower: *Number of classified and un-classified shapes from all three database, and the fraction of classified shapes with total shapes.*

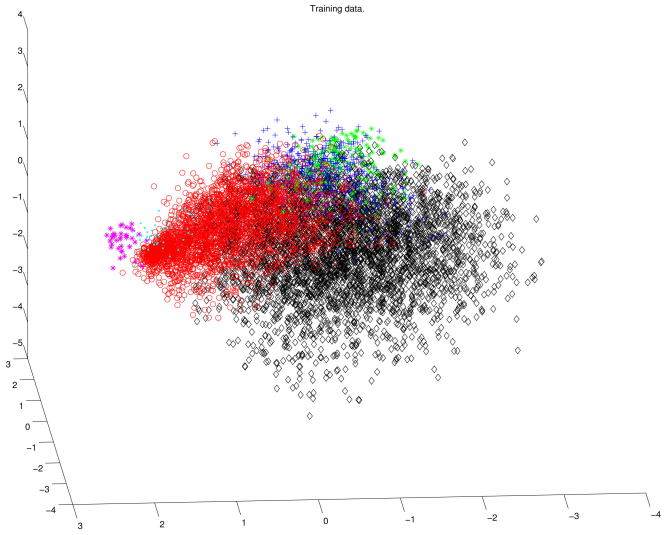
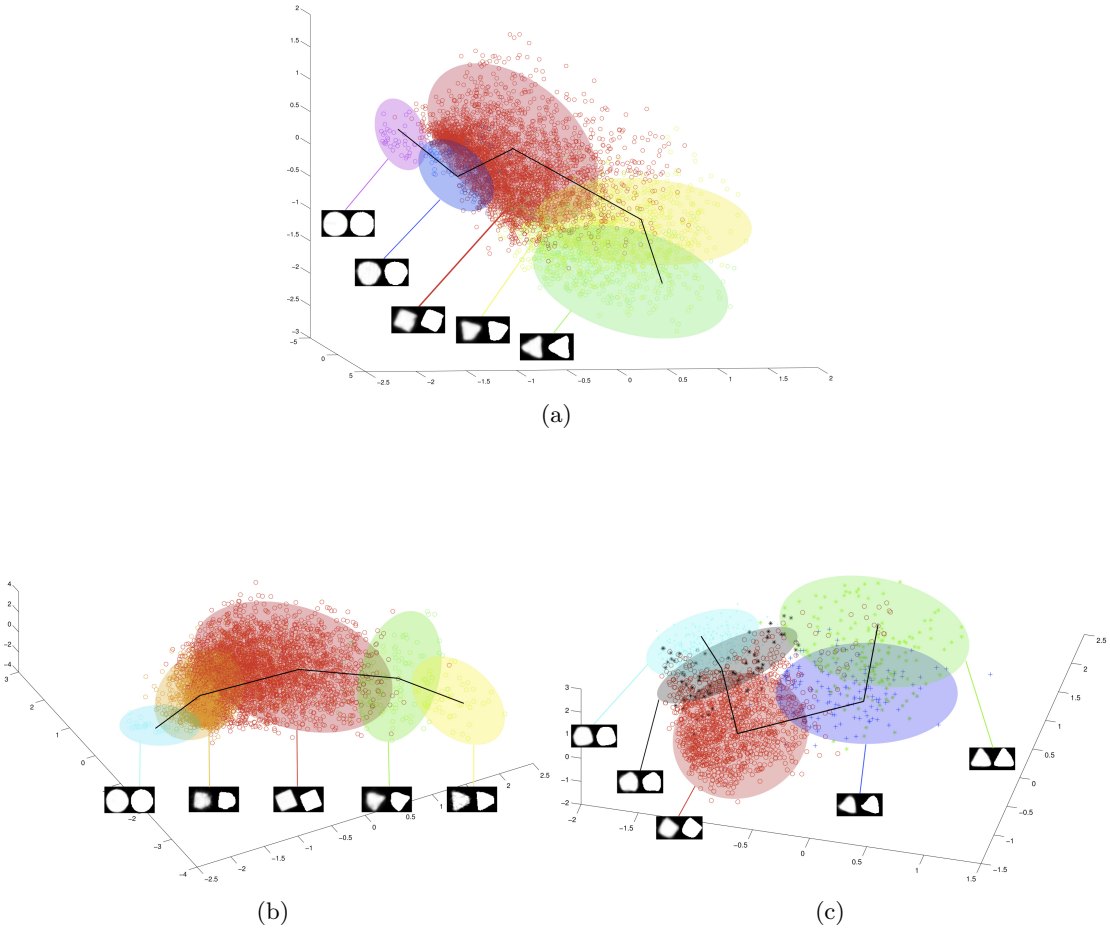


Figure 3-10: *All the data in the feature space. Features are the first three components of the PCA. Purple Star: Circles. Cyan Point: Polygons (R2C). Red Circles: Rectangles. Blue plus: Convex Quadrangle (R2T). Green Star: Triangles. Black Diamond: Random shapes*



(a)

(b)

(c)

Figure 3-11: *Distribution of primitive shapes' data in the feature space from MIT database by using different segmentation methods. (a) Thresholding (b) MSER (c) BSE.*

same feature space. As Figure 3-10 shows, the random shapes are apart from the primitive shapes classes. This suggests that primitive shapes are distinctive with the random shapes even in the signal world.

In the meantime, the distribution of primitive shapes' data in the feature space is also interesting and a regular pattern can be easily found. All the data of primitive shapes lie on a curve, as we drawn in Figure 3-11. Squares are always in the middle of feature space and circles in one side, while the triangles lie on the other side. Between these basic shapes, there are also some transitive shapes, which corresponding to blue and yellow part in the result, as shown in Figure 3-11.

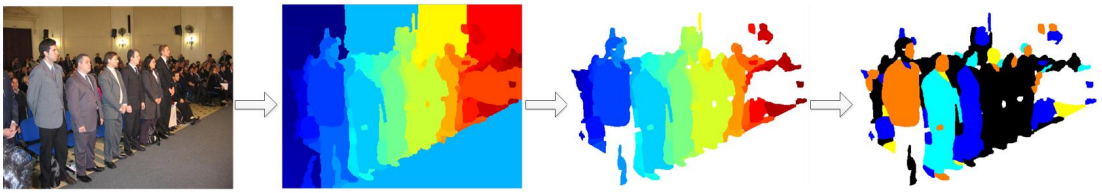


Figure 3-12: *An example of shape classification, from left to right: (a) original image. (b) segmentation (shape production). (c) any region that touched a picture boundary or which contained less than 100 pixels was removed from further consideration. (d) shape classification, the colourful parts are the regions classified as primitive shapes. The black parts are the regions can not be classified into any shape classes, known as random shapes.*

3.4 Application

3.4.1 Classify Regions into Primitive Shapes

Having found primitive shapes, we now make use of them. One obvious application is to classify regions in a new segmentation. To do this we construct a Gaussian mixture model (GMM) of the density of shape moments for each class that it outputs by the clustering algorithm. In addition, we construct a GMM over all the random shapes. Using random shape classes allows for the possibility that a given segmented region is not classified as a primitive shape. For a shape class S, let $\Omega = (N, \{\mu_i, C_i\})$ be a GMM with means μ_i and covariance matrices C_i, then $p(x|S) = \sum_{i=1}^{N} p(x|\Omega_i)p(\Omega_i)$. The posterior that shape x belongs to class S follows from Baye's rule

$$p(S|x) = \frac{p(x|S)p(S)}{\sum_{T \in \mathcal{S} \cup R} p(x|T)p(T)} \tag{3.22}$$

where $\mathcal{S}$ is the set of primitive shape class, and $\mathcal{R}$ is the random shape class. The priors $p(T)$ are the proportion of shapes clustered into class T from the image database being used; typically $p(T \in R) \approx 0.5$. Typical output can be seen in Figure 1-9 and Figure 3-12.

3.4.2 Scene Classification

We noticed that the priors on different primitive shapes depend on the database used, and these contain different sorts of photograph. The MIT database, for example contains street scenes, where as our natural database is exclusively landscapes, forests, coastal scenes *etc.* This suggests a scene classification application. Scene recognition is one of the most classic and challenging problems in computer vision. Many studies have presented approaches to classify indoor versus outdoor, urban versus country, sunset versus forest etc using global cues (e.g. power spectrum, color histogram information)[132, 98]. Oliva and Torralba [104] then proposed the idea that using

(a) Coast (b) Forest (c) Highway (d) Inside City

(e) Mountain (f) Open Country (g) Street (h) Tall Building

Figure 3-13: *Typical pictures from MIT Scene Classification database [104].*

global frequency with local spatial constraints (known as Spatial Envelope) to represent scenes, which are labelled with respect to local and global properties by human observation.

Our classifiers learn the ratio of prime shapes (and random shapes) associated with a given category of scene. We assume the ratio of priors for a given category is Dirichlet distributed, because the ratios for any given image sum to unity and are a multinomial distribution. Suppose $h1$ is the number of circles, $h2$ is the number of squares, the $p_i = h_i/sum_i$ is the multinomial. For each class we have many histograms and each histogram in a same class is little different but it is distinguished from those histograms that are not in the same class. Now we can treat each class as a collection of histograms, which is a distribution of multinomials. Dirichlet process controls the distribution of multinomials using just a few parameters, which can be find by fitting. And when a new histogram comes in, the dirichlet process will give the probability which class it belongs to. That is, if z is the ratio of priors then

$$p(z) \sim D(\beta_1, ..., \beta_K) = \frac{\Gamma(\Sigma_k \beta_k)}{\prod_k^K \Gamma(\beta_k)} \prod_k^K z_k^{\beta_k - 1}, \tag{3.23}$$

where Γ denotes the gamma function, and β are the Dirichlet parameters found by fitting [99]. Each distinct scene category, C has a distinct vector of β values. Given a new scene it is then easy to compute its ratio of prime shapes (and random), z, and hence compute $p(z|C)$ and therefore the posterior $p(C|z)$.

We used MIT scene classification database [104], partitioning the data into 800

	T	I	S	H	C	O	M	F
tal	**80**	0	0	11	0	0	0	9
ins	0	**85**	3	5	0	5	2	0
str	2	**42**	**20**	15	3	0	17	1
hig	7	1	9	**83**	0	0	0	0
coa	3	0	3	19	**75**	0	0	0
ope	17	0	4	0	17	**2**	**47**	13
mou	1	0	0	0	9	0	**90**	0
for	11	0	1	0	4	1	31	**52**

	T	I	S	H	C	O	M	F
tal	**82**	9	2	0	0	0	5	1
ins	3	**90**	3	1	0	1	0	0
str	1	5	**89**	2	0	1	2	1
hig	0	3	2	**87**	4	4	1	0
coa	0	0	0	8	**79**	12	1	0
ope	0	0	2	5	13	**71**	6	3
mou	1	0	2	2	2	5	**81**	7
for	1	0	0	0	0	1	6	**91**

Table 3.4: *Confusion Matrix.(Top):Our Proposed Method (Green: Urban Scene, Yellow: Natural Scene),(Bottom):Spatial Envelope [104]*

training images and 800 test images. The test set images have given ground-truth categories, so we could produce the confusion matrix seen in Table 3.4 — next to results from [104] for comparison, which is representative of state of the art. Our result is not quite as strong as [104], but strong nonetheless; broad classes such as 'Urban' and 'Natural' are very well classified. Given our approach uses much less information and is a simpler algorithm than any state of the art alternative, we found this to be a surprising result. In our results, most of the 'Street' scenes are classified as 'Inside City' because the ratios of primitive shapes between these two classes are very similar - there are many rectangles (squares) in these kinds of scene, such as buildings, windows, doors etc. This leads to a dramatically decrease of 'Street' category. The same case can be observed between 'Open Country' and 'Mountain' scenes - there are more 'triangles' and random shapes share between these two categories. Thus most of the 'Open Country' confused with the 'Mountain' class. However, the subject of this chapter is not scene classification and we are not motivated here to add more information that will bring us to match or exceed state of the art.

3.5 Limitation and Discission

In this work, we have tried our best to avoid introducing human interactions, in order to proving that primitive shapes such as square, triangle, circle etc emerge naturally from

images. However, in the practice, several parameters have to be introduced due the technical limitation. For example, the moment order n in equation 3.7 and bandwidth for mean-shift clustering in equation 3.13.

The perspective effects on the shape regions is another issue we ignored in this work. However, after some investigation on the segmented regions in original images, we find that those regions who are classified as 'Trapezium' are actually rectangles but observed in other perspectives. This means some of the shapes in 'Trapezium' are misclassified. They actually belong to the 'Square' class. We believe this problem is caused by the descriptor we are using, the moment, which is only scale/rotation invariant, but not perspective projection invariant.

Moreover, other than the primitive shapes we discussed in this work, we believe that edges, contours and curves might form part of a depiction basis, and these features should be used in the cross-depiction object modelling process. This is not only because there are large amount of artworks are depicted as line drawings, but also there are many objects and scenes only can be described by using edges and curve features, such as waves, winds and so on. There are many works focusing on using edges to model object classes across different depictive styles. For example, in [18], Cao et al proposed a novel index structure and the corresponding raw contour-based matching algorithm to calculate the similarity between a sketch query and natural images, and make sketch-based image retrieval scalable to millions of images. This method also works on cartoon and clip art, whose contours features are very distinct and clear. Hence, combing different local features such as edges, shapes, contours to model visual object class across different depictive styles is a potential direction.

3.6 Conclusion

The discovery of this work is unique, so far as we know: regions in image segmentations naturally form classes that correspond to simple, easily recognisable shapes, upon given appropriate shape descriptors and well-designed classifiers.

Clustering and other details such as alignment and noise handling can no doubt be improved, perhaps to sharpen the output icons. Also, we may want to consider for perspective projection, meaning creating shapes under a homography rather than an affine transform. Yet the results clearly show primitive shapes emerging from segmentations: they are 'features in the signal', and as such may be of use to many applications in computer vision and maybe elsewhere, not just scene and object classification.

In the next chapter, we present an application of modelling visual object class regardless of depictive styles based on primitive shapes we found in this research, go with a hierarchical graph model.

CHAPTER 4

A HIERARCHICAL GRAPH DESCRIPTION OF OBJECT CLASSES

4.1 Introduction

The fact that objects can be visualised in a wide variety of depictive styles, yet remain recognisable, leads us to the question: *what properties of an object class are invariant to depiction?* This is an important question for Computer Vision, because it directly affects performance in applications such as image retrieval, image matching, and object classification. With few exceptions, the models used in Computer Vision are trained and tested on a single depictive style. Yet models learned exclusively from photographs typically do not generalise well to other depictive styles; it can be said that such models are over-fitted. Such models are necessarily limited in their utility to applications – it becomes difficult to access both photographs and artwork in a library of portraits, for example. Additionally, recognisable objects exist for which there are no photographs (*eg* the Gryphon).

We argue that models of visual objects should not be premised, even tacitly, on photo-real appearance or indeed on any particular depictive style at all. Rather, visual object models should be based on quasi-invariant properties of the objects in a class. A similar argument is made by those who advocate part-based representations for image. We go further by saying that such models should generalise across depictive styles. This means that if a model is constructed using images in one style, the same object should also be classifiable even when depicted using a different style. This is also the main target of this thesis.

In the previous chapter, we have empirically show that the collection of 'primitive shapes' is a common property existed in both photos and art works, and they can be used as features to represent image. In this chapter, we investigate a method for

Figure 4-1: *The corner of an eye. The variance in appearance over photographs may be small enough to warrant the construction of a visual word, but a corresponding feature drawn from artwork may not lie within the cluster around that "photographic" word, due to the wide variation.*

modelling visual objects classes in a manner that is invariant to depictive style. The assumption we make is that an object class is characterised by the qualitative shape of object parts and their structural arrangement. Hence we use a graph of nodes and arcs in which qualitative shapes such as triangle, square, and circle to label the nodes. More exactly our model is a hierarchy of levels, yielding a coarse-to-fine representation. Each level contains an undirected graph of nodes and arcs. Nodes between levels are connected via parent-child arcs, which are directed. Child nodes are nested inside their parent.

In the Chapter 2, we have reviewed the bag-of-words family, one of the most popular visual object class modelling methods. Although the BoW methods address many difficult issues, they tend to generalise poorly across depictive styles. This means that models trained on photographs will tend to misclassify objects in another depictive style. The explanation for this is the formation of visual words, which are typically identified by clustering feature vectors that describe the appearance of image patches (*e.g.* SURF, SIFT). These feature vectors are designed to be robust to variations in lighting, affine transforms, colour changes *etc.* Finding visual words by clustering implies a tacit assumption: a sufficiently narrow variance in the appearance. This assumption explains why BoW models do not generalise well in cross-style problems (see our results in section 4.3 for direct empirical evidence). Consider, for example, a semantic feature such as the corner of an eye (see figure 4-1). The variance in appearance over photographs may be small enough to warrant the construction of a visual word, but a corresponding feature drawn from artwork may not lie within the cluster around that "photographic" word. This problem has been acknowledged by others, who respond by using geometric based features as words: Gu *et al.* use region shape [66]; Shotton *at al* use edgelets [123] as do Ferrari *et al.* [51]. Even so, many works of art remain beyond these classifiers. We do not use edge data, but do use region shape. However, rather than using complicated shapes for regions (as others do), or just using (a hierarchy of) Gaussian blobs [122], we use a collection of primitive shapes

(*eg* circle, square, triangle). The idea is that abstracting region shape into one of a few classes brings greater robustness to non-salient variations. Anecdotal support for this is found in the fact that many artworks comprise simple shapes, and even sophisticated artists often paint over a skeleton comprising simple shapes.

Our model is a hierarchical graph, in which simple shapes label nodes. We are not alone in using a parts based hierarchy to model objects and object classes. Hierarchy of shapes or object regions are used to learn object class models , for example [2, 46]. These build object class models, and most are motivated by a view we share: that such models should reflect the underlying object rather than its appearance. Many hierarchies make use of spatial data [106, 102], as we do by labelling arcs with displacement vectors. None of the above use a median graph, as we do, to represent a visual object class. We construct a median graph via embedding [52]. Others construct a class specific *graph prototype* [165], but this is not the median graph and is labelled with SIFT features rather than qualitative shape.

Contributions

Work presented in this chapter has been published in BMVC 2013 [164].

Our technical contribution is to show that *it is possible to learn models of object classes that generalise across depictive styles, in the sense that it is possible to learn a model using one style but classify objects depicted in other styles.* The chapter has two main sections:

- Section 4.2 explains how to build a hierarchical graph model to represent object classes, with nodes labelled by qualitative shape and edges labelled with displacement vectors.
- Section 4.3 describes experiments on a cross-depiction image dataset. The experiments provide empirical evidence that our model is more robust to cross-depiction object classification than an excellent Bag of Words classifier.

The paper concludes, in Section 4.5, with a discussion of the limitations of our modelling scheme, and points to future developments and applications.

4.2 Learning Model

We learn visual class models from input images, each labelled with the object they contain. There are three major steps: (i) build an "image graph" for each image in the training set; (ii) compute the class model as the median graph of the image graphs, and (iii) refine the class model by maximising classification performance over the training set. Figure 4-2 presents a framework of the proposed method. The steps are now discussed in detail.

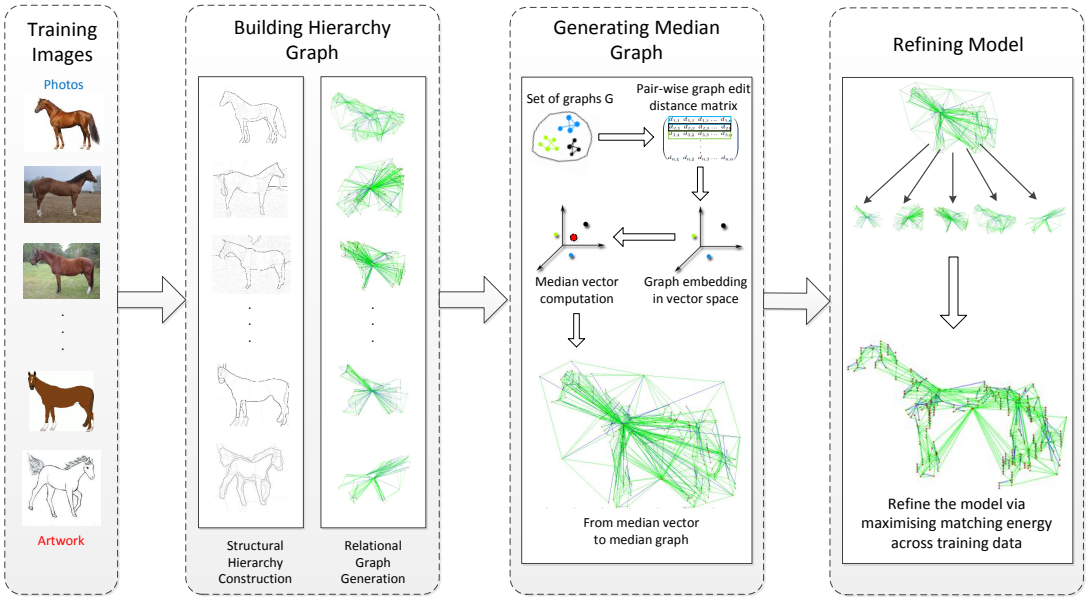


Figure 4-2: *Constructing a class model, from left to right. (a): An input collection (possibly different depictions) used for training. (b): Probability maps for each input image, and graph models for each map. (c): The median graph model for the whole class. (d): The refined median graph as the final class model.*

4.2.1 Build Image Graphs, one for each image.

Our modeler uses a state of the art segmentation algorithm from Berkeley that automatically yields a hierarchical description of an input image at first.

Given an image, the well known gPb (Global Probabilities at Boundaries) [94] is used to obtain the contour signal $S(x, y, \theta)$, which predicts the probability of an image boundary at location (x, y) and orientation θ. Based on the contour signal input, the hierarchical regions are constructed after performing two transformations, the Oriented Watershed Transform (OWT) and Ultrametric Contour Map (UCM). In order to compute the OWT of an image, watershed arcs are then approximated to line segments, whose slopes determine orientations between neighbouring regions. At the same time, consistency of contours is maintained by ensuring that only the maximal contour response over the space of all orientations is retained at every pixel. Afterwards, the UCM is computed by weighting every contour in the transformed image, according to the similarity between interesting edges of the regions separated by the contour. Given a sequence of contour strengths, the region hierarchy is constructed by a greedy graph-based region merging algorithm which will merge the most similar regions. This outputs a sequence of segmentations indexed by thresholding over a probability map over region boundaries. The segmentations are ordered coarse-to-fine; smaller regions are nested inside larger ones.

However, one disadvantage of the hierarchies produced by the above algorithm is the large layer size. It contains far more data than it is required for an efficient descriptor, which is necessary for the further graph matching process. Their number can be reduced to about ten or so, without loss of information, by a graph based filtering process [128]. The reduced hierarchies preserve the semantic interpretation in terms of objects and object parts; the number of levels of the reduced hierarchies are typically an order of magnitude less than the original. The principle in solving the problem of filtering hierarchy is to choose those levels that are lower in complexity than their neighbors. The Laplacian graph energy is used to measure the complexity. Let G be a graph of n vertices and m edges. Let A and D be its adjacency matrix and corresponding degree matrix. Then $L = D - A$ is the graph Laplacian. The Laplacian graph energy is defined by

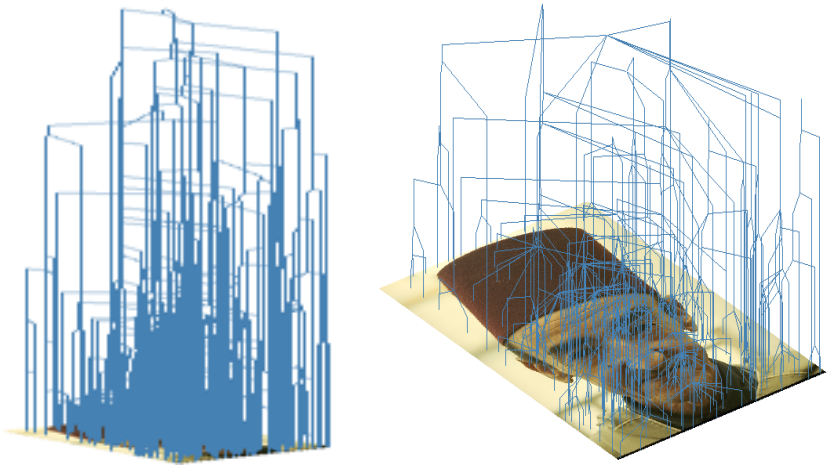
$$\mathscr{LE}(G) = \sum_{j=1}^{n} |\mu_j - 2m/n| \tag{4.1}$$

where μ_j is the eigenvalues of L and $2m/n$ is the average vertex degree. In [128], the affinity matrix is specially defined as $A = \{a_{ij} | a_{ij} = \exp(-w_{ij}/w_{max})\}$ where w_{ij} is the average boundary strength between regions i and j, and w_{max} is a decay factor set to the maximum over all w_{ij}. Another extension, called the component-wise Laplacian graph energy is introduced in [128]. For a graph with k disconnected components, the cLGE is defined as

$$c\mathscr{LE}(G) = K \sum_{i=1}^{k} \frac{\mathscr{LE}(G_i)}{|n_i|} \tag{4.2}$$

in which G_i is the i th connected component (or sub-graph) of $|n_i|$ nodes, and K is the number of nodes in the whole graph. The cLGE at every level in the hierarchy is computed independently using graphs built from the primitives at the lowest level. At the bottom level of the hierarchy, each primitive is an 1-node subgraph on its own, whereas the top level forms a single connected graph. At intermediate levels, as segmentations become coarser, subgraphs are merged to create larger ones, and so the number of disconnected components will fall. cLGE for the level as a whole can rise or fall, depending on the way these primitives are connected. So only those levels, at which cLGE is locally minimal, are kept in the filtered hierarchy [128]. Figure 4-3 displays the comparison between the gPb-owt-ucm tree and the hierarchies after filtering by method proposed in [128].

After the structural hierarchies construction, we transfer them to a attributed relational graph. Building a graph $G =< V, E >$ from the segmentation hierarchy is straight-forward. Each segmented region in the hierarchy is represented as a vertex(node) $v \in V$, where V is the set of all vertices. The set of edges E contains edges $\{e_{ij} = v_i v_j | v_i, v_j \in V\}$ that connect vertices v_i and v_j. In our case, v_i and v_j



(a) (b)

Figure 4-3: (a) *All levels of a gPb-owt-ucm hierarchy.*(b) *The levels remain after filtering [128]*

belong to the same level, or neighbouring levels of the hierarchy. Nodes at the same level are connected by an edge if their corresponding regions share a boundary in the segmented regions (the black segments in the figure 4-4). Parent-children connectivity is decided by checking regions that intersect across two neighbouring layers of the hierarchy (the colorful segments in the figure 4-4). Graph G now contains an efficient representation of the spatial arrangement of segmented regions, and encapsulates relations of adjacency and containment between regions. This graph is our starting point. Typical examples of graph models can be seen in Figure 4-5.

We label graph nodes with qualitative shape, that is shape class, and arcs with relative displacement vectors. The displacement vectors link the centroid of one region

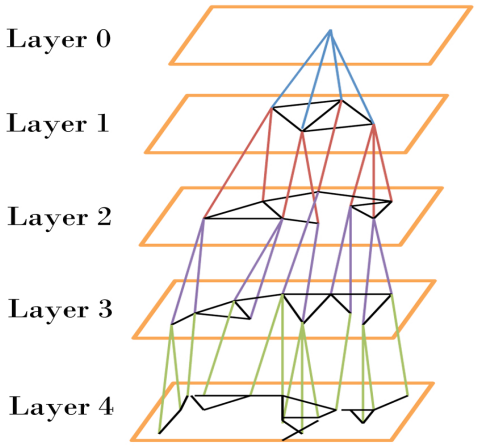
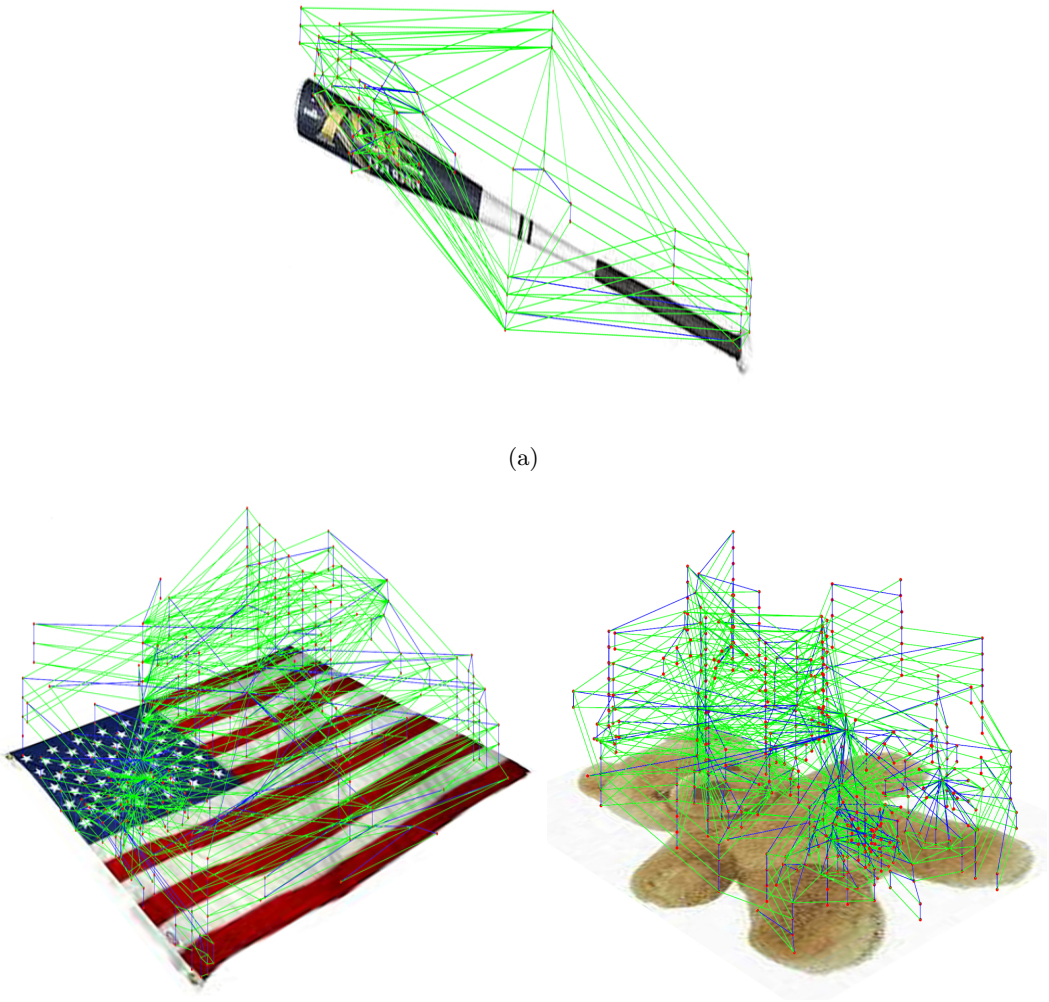


Figure 4-4: *Relational graph model in schematic form.*

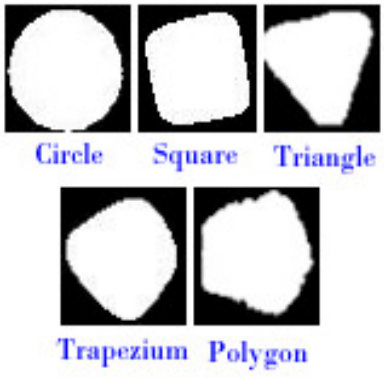


(a) (b) (c)

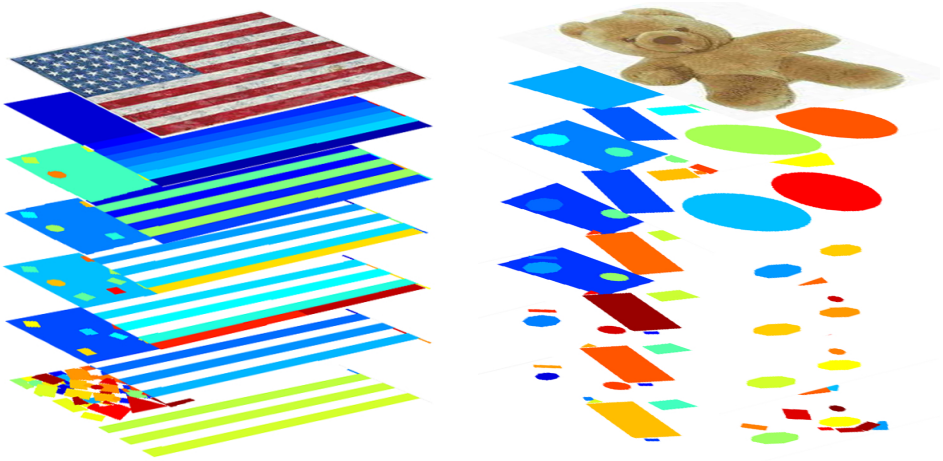
Figure 4-5: *A graph model* (a) *a bat* (b) *an American flag* (c) *a teddy bear. Parent-child arcs in blue, neighbour arcs in green.*

to the centroid of a neighbour, so are easy to compute. Qualitative shape is a class label ($\mathbb{S} = \{circle, polygon, square, trapezium, triangle, random\}$) found by directly classifying the region obtained from segmentation based on clustering the density distribution of (the absolute value of) Zernike moments. The classifier explicitly model the density of regions that do not belong to one of the named classes, we call this class *random*; it is the sixth of the shape classes we use. See 3.2 for details on shape classification. "Polygon" captures pentagons, hexagons, etc.

More exactly, we label nodes with probability vectors over $\mathbb{S}$. The shape classifier is a mixture model over a feature space of (the absolute value of) Zernike moments. Each mixture component is itself a Gaussian Mixture Model. For each shape class $S \in \mathbb{S}$ we specify a GMM $(N_S, \{\alpha_{si}, \mu_{Si}, C_{Si}\}_{i=1}^{N})$, with N_S the number of GMM components,



(a)



(b) (c)

Figure 4-6: (a) *Primitive shape classes (other than* random*)* (b) *An American flag broken in primitive shapes.* (c) *A teddy bear likewise decomposed.*

and α_{Si}, μ_{Si}, C_{Si} being the prior, mean, and covariance of each. For a region x we denote the density of the shape class by $p(x|S)$, which is readily computed using the standard form for a GMM,

$$p(x|S) = \sum_{i=1}^{N_S} p(x|\mu_{Si}, C_{Si})\alpha_{Si}. \tag{4.3}$$

We label the corresponding graph node with a 6 elements-vector of MAP estimate of shape-class membership:

$$p(S|x) = \frac{p(x|S)p(S)}{\sum_{T \in \mathcal{S}} p(x|T)p(T)}. \tag{4.4}$$

If an application requires a single shape, we use $S^* = \arg\max_S p(x|S)$. The shape-class prior, $p(S)$ is taken to be the relative number of shapes classified as shape S.

All parameters used are provided by the shape classifier after training on about 40000 regions. Figure 4-6 illustrates the shape classes we use, and the shape classes used to label nodes at each level of a hierarchy. This completes our construction of an image graph.

4.2.2 Compute an Initial Visual Class Model.

Given a set of image graphs, the next step is to compute the median graph model as the visual class model. The median graph, introduced into structural pattern recognition by Jiang etal [78], is a useful concept that can be used to represent a set of graphs. A single prototype is extracted from a collection of graphs.

Let $\mathbb{G} = \{G_1, ...G_n\}$ be a set of graphs and let $d(G_i, G_j)$ be some distance function to measure the dissimilarity between graphs G_i and G_j. A simple approach to finding a median graph is to find the graph $G_k \in \mathbb{G}$ that minimises the sum of $d(.,.)$ over $\mathbb{G}$. A better approach is to choose the median graph, $\bar{G}$ from the set of all graphs that can be constructed from all combinations of all subgraphs of all graphs $G \in \mathbb{G}$. This vast set is denoted $\mathbb{U}$, and the median graph we use is defined using it:

$$\bar{G} = \arg\min_{\bar{G} \in \mathbb{U}} \sum_{G_i \in \mathbb{G}}^{n} d(\bar{G}, G_i). \tag{4.5}$$

This is far too large a problem to solve directly. In this paper we use an approximate algorithm for median graph computation proposed in [52].

For a set of image graphs generated as the section 4.2.1, $\mathbb{G} = \{G_1, G_2, ...G_n\}$, we first compute the graph edit distance (equal to the cost of a sequence of optimal edit operations, see section 4.2.2) between every pair of graphs in $\mathbb{G}$. Hence, an $n \times n$ distance matrix will be generated. Then, each row/column of the matrix can be seen as an n-dimensional vector, corresponding to each graph in $\mathbb{G}$. This embeds graphs into an n-dimensional feature space. Secondly, a median vector will be generated by computing the *Euclidean Median* of all the data points in the feature space. Finally, we transfer this median vector to a graph representation. This transformation process involves a *triangulation procedure*, more details can be found in the following section. The result is our first approximation of the visual class model.

Graph Edit Distance

The graph edit distance, $d(G_1, G_2)$, of two graphs is equal to the cost of an optimal *ecgm* (error-tolerant graph matching) [16]. Formally, let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ to be two graphs, and the number of vertices of two graphs is not necessary equal. An error-correcting graph matching (*ecgm*) from G_1 to G_2 is a bijective mapping $X : \hat{v}_1 \leftrightarrow \hat{v}_2$, where $\hat{v}_1 \in V_1$ and $\hat{v}_2 \in V_2$, so the number of vertices of two matched sub-graph

$|\hat{v}_1| = |\hat{v}_2|$. Then,

$$d(G_1, G_2) = c(X^*) \tag{4.6}$$

The cost function $c(X^*)$ here is the sum of distance of the edit operations implied by X^*, which is the optimal *ecgm* mapping and can be obtained by a global optimization process. The mapping X^* directly implies an edit operation on each node in G_1 and G_2. For example, we say that node $x \in \hat{v}_1$ is substituted by node $y \in \hat{v}_2$ if $X^*_{xy} = 1$. Any node from $V_1 - \hat{v}_1$ is deleted from G_1, and any node from $V_2 - \hat{v}_2$ is inserted in G_2 under mapping X^*. Additionally, the mapping X^* indirectly implies edit operations on the edge of G_1 and G_2.

We formulate the mapping process into a graph matching scheme. Given a pair of graphs, G_1 and G_2, the graph matching problem consists in finding a correspondence between nodes of G_1 and G_2 that maximise the following score of global consistency given as

$$E(X; G_1, G_2) = \sum_{i \in V_1, j \in V_2} x_{ij} \Phi_{i,j} + \sum_{i_1, i_2 \in V_1, j_1, j_2 \in V_2} x_{i_1 j_1} x_{i_2 j_2} \Theta_{e_1 e_2}, \tag{4.7}$$

where each X is a binary matrix that denotes the node-node correspondence and $e_1 = (i_1, i_2) \in E_1$, $e_2 = (j_1, j_2) \in E_2$. Maximising E gives an optimal edit path between two graphs, X^*. The two similarity matrices, Φ and Θ, measure the similarity of each node and each pair of edge respectively. We follow Torresani et al [140], who proposed a dual decomposition approach to solve this NP-hard programming in Eq.4.7.

Then we can let the graph edit distance

$$d(G1, G2) = c(X^*) = \exp(-E(X; G_1, G_2)) \tag{4.8}$$

We specify Φ as the probability that two segmented regions are the same underlying simple shape. Suppose we have region i in graph G_1 and region j in G_2, then $p(S|i, j)$ denotes the probability that both are simple shape S. We specify

$$\Phi_{ij} := \max_{S \in \mathbb{S}} p(S|i, j) = \max_{S \in \mathbb{S}} p(S|i) p(S|j), \tag{4.9}$$

which assumes that regions are iid. This makes it easy to compute Φ_{ij} via equation 4.4.

The similarity of a pair of edges from two graphs, Θ, is obtained by evaluating how well the edge e_1 in graph G_1 matches the edge e_2 in graph G_2, in terms of both length and direction. Following [140] we specify edge similarity as

$$\Theta_{e_1 e_2} := \exp(\eta(1 - \exp(\delta^2_{e_1 e_2}/\sigma^2_\iota)) + (1 - \eta)(1 - \exp(\alpha^2_{e_1 e_2}/\sigma^2_\alpha))) \tag{4.10}$$

in which, using p_1, p_2 and q_1, q_2 to denote region centroid of i_1, j_1, i_2, j_2:

$$\delta_{e1,e2} = \frac{|||p_1 - q_1|| - ||p_2 - q_2|||}{||p_1 - q_1|| + ||p_2 - q_2||} \text{ and } \alpha_{e1,e2} = \arccos\left(\frac{p_1 - q_1}{||p_1 - q_1||} \cdot \frac{p_2 - q_2}{||p_2 - q_2||}\right). \quad (4.11)$$

The parameter η is a scalar value trading off the importance of preserving distance versus preserving directions, we set $\eta = 0.5$. Variance values σ_ι^2 and σ_α^2 could (in principle) be learned from ground truth correspondences, but we set $\sigma_\iota^2 = 0.5$ and $\sigma_\alpha^2 = 0.9$ as the initialized value given by [140].

Median Graph Generation

Given a set of graphs $\mathbb{G}$, we compute the graph edit distance between every pair using above equations. These distances are arranged in a distance matrix. Each row/column of the matrix can be seen as an n-dimensional vector. Since each row/column of the distance matrix is assigned to one graph, such an n-dimensional vector is the vectorial representation of the corresponding graph. Once all the graphs have been embedded in the vector space, the median vector is computed, using Euclidean Median.

$$Euclidean\ median = \arg\min_{y \in \mathbb{R}^n} \sum_{i=1}^{m} ||x_i - y|| \quad (4.12)$$

where $||x_i - y||$ denoted the Euclidean distance between the points $x_i, y \in \mathbb{R}^n$. The Euclidean Median, is a point $y \in \mathbb{R}^n$ that minimizes the sum of the Euclidean distance to all the points in X.

To approximate a median graph, we employ a triangulation procedure, illustrated in figure 4-7, works as follows. Given the n-dimensional points representing every graph in $\mathbb{G}$ (the white dots in figure 4-7 (a)) and the Euclidean Median vector v_m (the grey dots in figure 4-7 (a)) computed in the last step, we first select the three closest points to the Euclidean median (v_1 to v_3 in figure 4-7 (a)). Notice that we know the corresponding graph of each these points. Then, we compute the median vector v'_m of these three points (the black dot in figure 4-7 (a)). v'_m is in the plane formed by v_1, v_2 and v_3. With v_1 to v_3 and v'_m at hand (figure 4-7 (b)), we arbitrarily choose two out of three points (without loss of generality we can assume that we select v_1 and v_2) and we project the remaining point (v_3) onto the line joining v_1 and v_2. In this way, we obtain a point v_i in between v_1 and v_2 (figure 4-7 (c)). With this point at hand, we can compute the percentage of the distance in between v_1 and v_2 where v_i is located (figure 4-7 (d)). As we know the corresponding graph of the points, we can obtain the graph g_i corresponding to v_i by applying the weighted mean procedure [17]. Once g_i is known, then we can obtain the percentage of distance in between v_i and v_3 where v'_m is located and obtain g'_m applying again the weighted mean procedure(figure 4-7 (f)). Finally, g'_m is chosen as the approximation for the generalized median of the set $\mathbb{G}$.

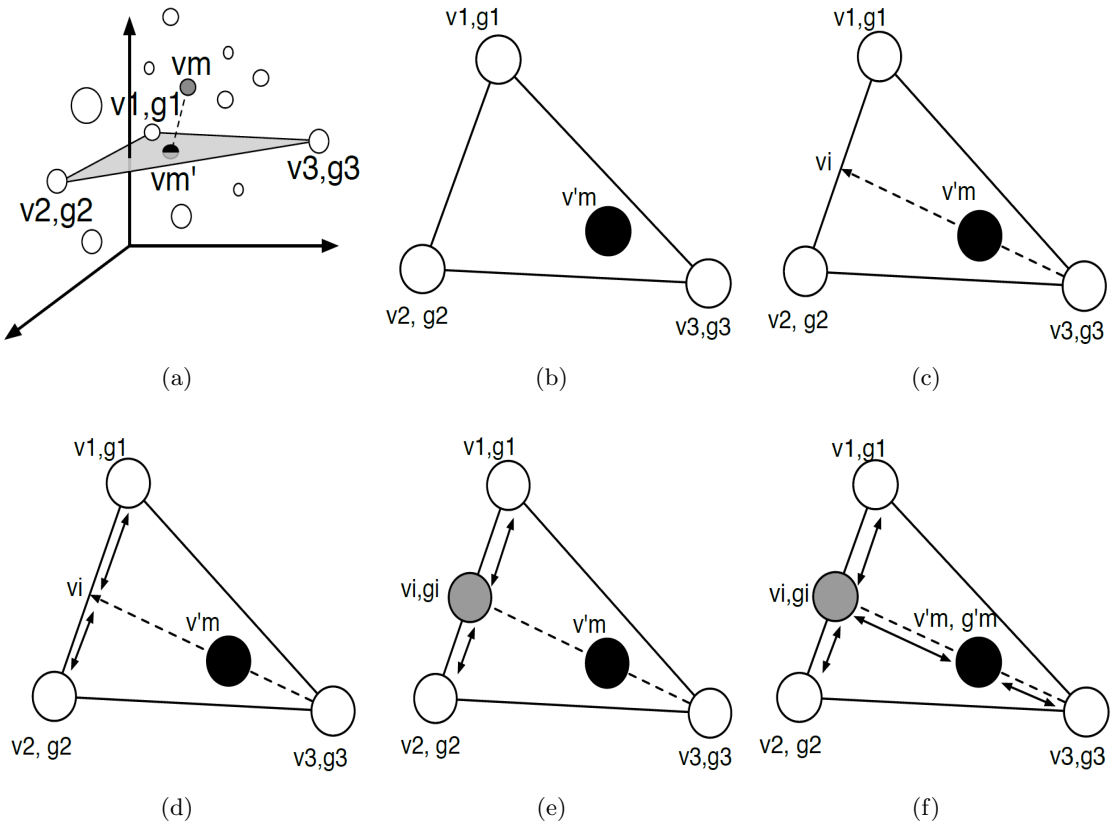


Figure 4-7: *Triangulation Procedure [52]: (a): Given the n-dimensional points representing every graph in $\mathbb{G}$ and the Euclidean Median vector v_m, we first select the three closest points to the Euclidean median. Then, we compute the median vector v'_m of these three points. v'_m is in the plane formed by v_1, v_2 and v_3. (b): With v_1 to v_3 and v'_m at hand. (c): we arbitrarily choose two out of three points and we project the remaining point (v_3) onto the line joining v_1 and v_2. In this way, we obtain a point v_i in between v_1 and v_2. (d): With this point at hand, we can compute the percentage of the distance in between v_1 and v_2 where v_i is located. (e): As we know the corresponding graph of the points, we can obtain the graph g_i corresponding to v_i by applying the weighted mean procedure [17]. (f): Once g_i is known, then we can obtain the percentage of distance in between v_i and v_3 where v'_m is located and obtain g'_m applying again the weighted mean procedure. Finally, g'_m is chosen as the approximation for the generalized median of the set $\mathbb{G}$.*

4.2.3 Refine the Visual Class Model.

The median graph contains nodes and arcs that derive from visual clutter in background of images in the training set. Hence, we developed a cleaning algorithm to remove such elements, and so refine the visual class model (*vcm*).

We begin by matching the median graph back into each training image, to count the number of times a given node in the model appears in the training data. This frequency count indicates the relevance of a node to the visual class. Next, we delete all nodes below a frequency threshold – we compute the matching score (using equation 4.7) between the edited *vcm* and each image in the training set. The threshold is then

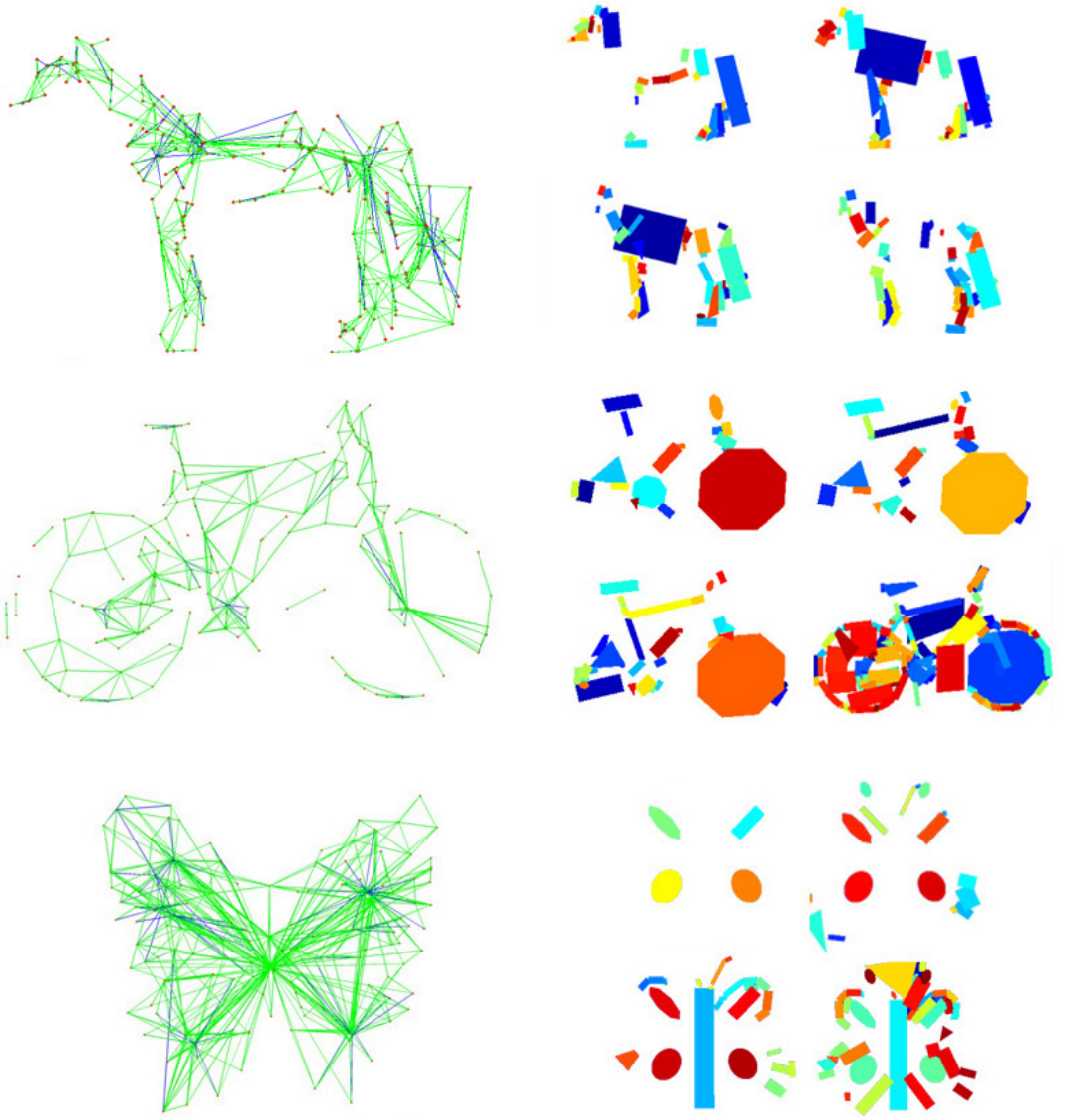


Figure 4-8: *Examples of three graph models generated from 3 categories of objects, which are horses, bicycles, and butterflies. The visualization shows of selected levels below the corresponding model, with the simple shapes fitted. Child-parent arcs are in blue, adjacencies between the nodes in the same level are green.*

incremented, and the process repeats until the total match score is maximised. The nodes that remain define the final *vcm*. Figure 4-8 shows some final results.

4.3 Experiments and Results

Our visual class model (*vcm*) has the potential to be used in many applications, here we use classification – and cross-depiction classification in particular. Like any classification task, ours consists of two main steps, training and testing. Training comprises building a *vcm*, as described in Section 4.2. The testing process involves matching

Figure 4-9: *Some example pictures from our own dataset that augments CalTech 256.*

each *vcm* (one for each visual class under consideration) into the image graph that corresponds to an input test image. More formally, for n categories of object, we have a set of *vcms* with index set $\mathbb{N} = \{i\}_1^n$. Given an input image I we have its image graph $G[I]$, we compute

$$i = \arg\max_{i \in \mathbb{N}} E(X; G[I], G_i) \tag{4.13}$$

as the index number of the class to which the query image belongs. The similarity measure function is given by equation 4.7; notice that it ignores clutter nodes in $G[I]$.

To fulfil the cross-style object classification task, we augmented the Caltech-256 Object Category Dataset[64] with a parallel database that widens the variation in depictive style, see Figure 4-9. There are 20 categories of object this dataset.

Using our expanded version of CalTech-256 we conduct experiments designed to test how well a visual class model generalised from across depictive styles. Specifically we: (1i) train on photographs alone and test on photographs; (1ii) train on artwork alone and test on artwork; (2i) train on photographs alone and test on artwork; (2ii) train on artwork alone and test on photographs; (3i) train on photo and test on both photographs and artwork; (3ii) train on artwork and test on both photographs and artwork; (4) train on both and test on both.

For comparison with alternative visual class models we conduct the above experiment using not just our *vcm* but with two others also. The first is a BoW classifier, chosen because it performs well and will help us assess the performance of such a popular approach to the problem of cross-depiction classification. The BoW we use is proposed in [147], it uses PHOW features [15] (dense multi-SIFT descriptors) and K-means for visual word dictionary construction. Finally, it uses a linear SVM for classification. Here we set the number of words in vocabularies at 600. And the other parameters

are the same as [147] used, which can achieve 64% performance on Clatech101 dataset. The second *vcm* alternative we experiment with uses structure alone as a model [167] and is relevant because it explicitly sets out to classify in a cross-depiction domain. It uses the first few eigenvalues of the Laplacian matrix of the object structure as the feature vector, which embeds graphs in a pattern space. A GMM is employed as the classifier. We use our hierarchical graph structure as the input of this algorithm to compute its Laplacian matrix. Experimental results are shown in the following section.

4.3.1 Results and Discussion

Classification accuracy of different methods in various Training/Test cases, shown in table 4.1 (the deeper the color, the better the performance). The training and test images were selected to show objects on uncluttered backgrounds, which is also a limitation of this work. The numbers of images in the table are *per-class* figures, the rates are averaged over 20 classes. In total our test used 800 images, including our extension to CalTech 256.

case 1: Training	3a	5a	3p	5p
case 1: Testing	15a	15a	15p	15p
Dense SIFT [147]	57%	59%	66%	70%
Structure Only [167]	15%	19%	13%	16%
Proposed Method	59%	62%	60%	61%

case 2: Training	3p	5p	8p	10p	3a	5a	8a	10a
case 2: Testing	15a	15a	15a	15a	15p	15p	15p	15p
Dense SIFT [147]	34%	38%	43%	47%	35%	42%	49%	51%
Structure Only [167]	15%	15%	19%	23%	11%	15%	22%	25%
Proposed Method	52%	60%	63%	64%	56%	59%	64%	67%

case 3: Training	3a	5a	3p	5p
case 3: Testing	30m	30m	30m	30m
Dense SIFT [147]	46%	50%	50%	54%
Structure Only [167]	13%	16%	14%	16%
Proposed Method	58%	61%	56%	61%

case 4: Training	6m	10m
case 4: Testing	30m	30m
Dense SIFT [147]	60%	61%
Structure Only [167]	21%	24%
Proposed Method	62%	65%

Table 4.1: *Classification accuracy for different cases. From top to bottom: (case 1:) single depiction task. (case 2:) cross depiction task. (case 3:) single to mixture depiction task, and (case 4:) mixture to mixture task. The character 'p' is 'photos', 'a' is 'art' and 'm' is 'mixture'. The deeper the color, the better the performance.The numbers of images in the table are* per-class *figures, the rates are averaged over 20 classes. In total our test used 800 images, including our extension to CalTech 256.*

From the results, it is clear shown that our proposed method performs better than both traditional bag-of-words method and structure only method in terms of cross-

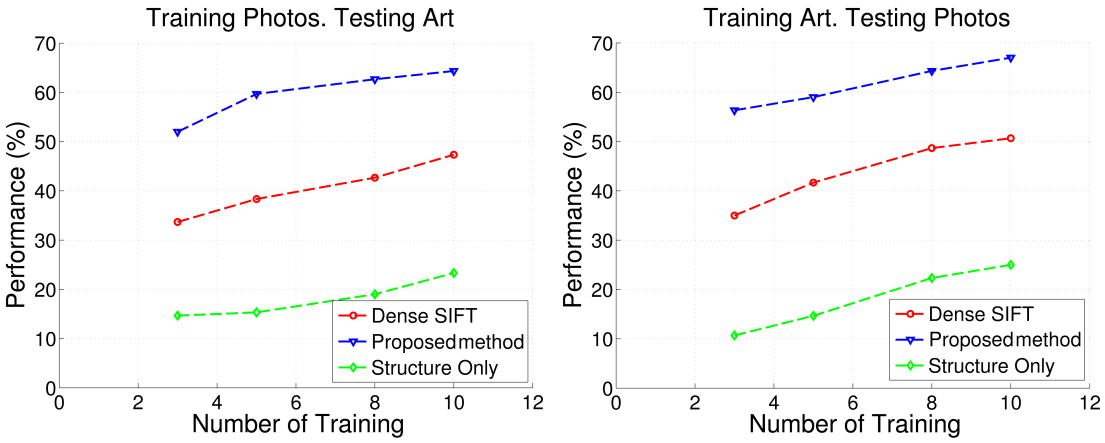


Figure 4-10: *Performance trend when using different numbers of training images in case 2, the single cross depiction task.* (a) *Photos to artwork classification task.* (b) *Artwork to photos classification task. Our method (the blue one) out performs the other two method obviously.*

depiction generalisation. This is most obvious when training in one depiction and testing in another: the accuracy of our method is nearly 20 percent points higher than the method using dense SIFT, and nearly 50 percentage points higher than structure alone. The traditional BoW method is superior by up to 9% when photographs are used for both training and testing – which is the kind of result we anticipated since BoW models are tuned to low-variance features whereas we set out to allow for wider variation. The low score of the structure-only method may be explained by our use of more complex structures than the original [167].

The case of training on both photographs and artwork is interesting. When photographs are the test case BoW and our method perform about equally, but our method performs the better when artwork is used to test. It is notable that "artwork" in fact covers a broad variety of depictions. This result suggests that word formation inside BoW is biased towards "photographic words", where we would expect the densest concentration of features. This is underlined by the fall in performance of BoW when the number of artworks in the training set rises. When all cases are taken into account, our method is much more stable in performance (from 52% to 67%) compared to BoW (34 % to 70%). Our rates compare favourably to CalTech 256 benchmarks using only photographs (see [64] and [147]). We are taking a first step towards widening the classification problem. Figure 4-10 show the performance trend when using different numbers of training images in case 2, the single cross depiction task, comparing with other two methods. And the confusion matrix for each test cases are provided in Appendix B.

Our system is implemented by matlab, running on a Core i7CPU 2.67GHz machine. The average training time for a single class is 3 hours. It takes such a long time due

to the pair-wise graph matching process during the median graph generation process. The average testing time of single time is 30 to 40 seconds.

4.4 Limitations

There are still some limitations of this current system, for example, the methods fail when object is relatively small with complicated background, because our method is highly relied on the segmented regions. Too crowd background would produce too many segmented regions, which will produce a highly complex candidate graph. And the objects have to be presented in canonical pose, for example, we only can accept the horse images which are captured in the side-view. The multi-view model is not considered in this work, but we popose a new modeling process which can handle multi-view instances in Chapter 5.

Moreover, the dataset is small, especially the number of positive examples. This is mainly due to the difficulty of collecting data. And we do not yet localise objects in images, such an ability would improve our ability to learn. Our class exemplars exhibit a complex structure that would benefit from further simplification, *eg* using *graph prototypes* rather than median graphs. Additional labelling (for example texture on nodes, and affine maps) may also improve classification performance. We cannot model objects that exhibit high variation in structure and/or shape, *eg* buildings as a general class, such broad classes are a challenge to many classifiers. Our method depends on matching and so can be slow, faster algorithms – perhaps via a hierarchy of classes – are desirable. Nonetheless, our results are a first step towards depiction invariant modelling.

4.5 Conclusion

The ability to generalise to new depictive styles is important, not least because the number of depictive styles is seemingly unbounded. No training procedure can capture them all and so a class model that is able to generalise to unseen depictive styles is of value. In this chapter , we proposed a hierarchical graph model, with qualitative shapes such as triangle, square, and circle to label the nodes. Experiments show that our proposal method performs better than the traditional visual appearance based method in cross-depiction problems (including to unseen depictive styles), in mixed problems, and in art-only problems.

With more depictive styles joined in and the image background became more complicated, how to capture the wide variation between different styles is still an open question. In the next chapter, we propose a cross-depiction visual object class modelling method to capture the wide variations, based on a more challenging cross-depiction

image dataset. Moreover, other that classification, we also use our proposed model in detection and recognition.

CHAPTER 5

LEARNING GRAPHS TO MODEL VISUAL OBJECT ACROSS DIFFERENT DEPICTIVE STYLES

5.1 Introduction

In the previous two chapters, we focused on the question of 'finding properties invariant to depiction of objects' and we have proven that a combination of global structural information and local regions (fitted by simple primitive shapes) could be useful in modelling objects regardless of different depictive styles. However, the limitation of this kind of representation is also obvious, for example, there is no striking structure can be detected in some object such as water, smoke and some modern buildings. Although these objects are out of the scope that we want to address in this thesis, as more depictive styles are included, the question of ' how to capture the wide variation in visual appearance exhibited by visual objects across depictive styles' is still needed to be answered.

Before pursing the answer, a more challenging public dataset is important for comparing current techniques, as this new area develops. We provide such a dataset in this chapter and we use our dataset to confirm by experiment the intuition that the cross-depiction problem is difficult because the variance across photo and art domains is much larger than either alone. We then extensively evaluate classification, domain adaptation and detection benchmarks for leading techniques, demonstrating that none perform consistently well given the cross-depiction problem.

Then, we provide a modelling scheme for visual class objects that generalises across a broad collection of depictive styles. Not like the primitive shape and graph based method we proposed in Chapter 4, in this work, the Primitive Shape features are dropped. Instead, we employed more powerful HoG features, which is also proved effective in cross-domain image matching [124]. The reason that the Primitive Shape

features are replaced is mainly because limitations we discussed in Section 4.4 have not been addressed, ie. the primitive shape producer are highly relied on image segmentations and how to screen out those regions only related to target objects from crowd background segmentations is difficult. In other words, although primitive shapes can highly abstract the object it does bring more noise than well-designed hand crafted features, such as SIFT and HoG. Moreover, since we are planning to use graph-based modelling method, but shape-level feature extractor produces too many nodes, which will lead to a too large graph. This causes large pressure for the further learning and mathcing.

The assumption we make is that a mixture model at the parts level - mixed within each part but depicted in different styles - can be used to characterise an object class of mixture depictive styles. Here, we propose to learn a model graph for each visual object class, based on a particular, but rather a general, graph representation, with histogram-based attributes for nodes and edges. Instead of using a single 'label' in each node of the graph, we use multi-labeled nodes to construct the graph model. These multi-labels corresponds to different depictive styles of each object part. Moreover, encouraged by [26] we use a max-margin framework to learn the weight for each part(node) and edge of our model graph to present different nodes and edges contributions, leading to better detection and matching results.

Contributions

Work presented in this chapter has been published in ECCV 2014 [162]. Our contributions of this chapter can be summarised as following:

1. We introduce a new photo-art dataset, *Photo-Art-50*, announced with bounding boxes, designed specifically for the cross-domain problem. Based on this new dataset, we evaluate leading recognition and detection techniques and two state-of-the-art domain adaptive methods for cross-depiction task (see section 5.2).

2. We introduce a new way in which we account for the wide variation in feature distributions, specifically - *the use of multi-labels to represent visual words that exists in possibly discontinuous regions of a feature space.* (see section 5.3)

3. We employ an SSVM method to learn the weights to encode the importance of nodes and edges similarities. (see section 5.3)

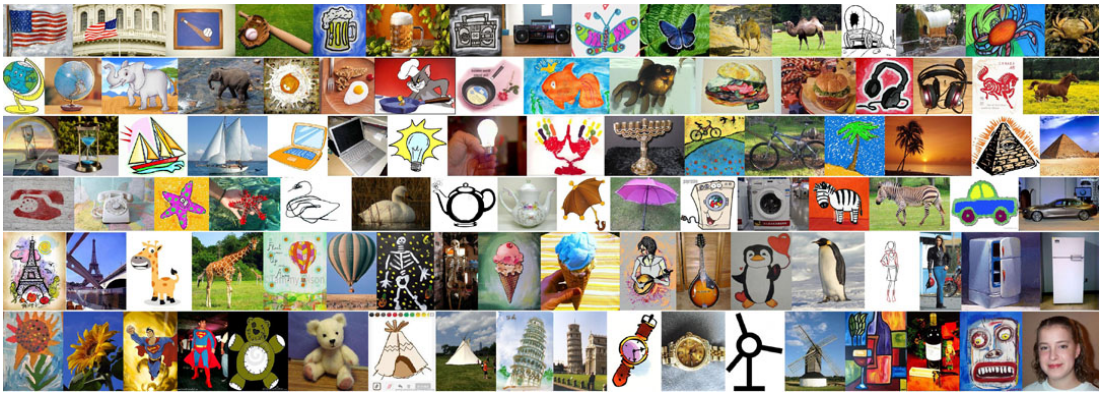


Figure 5-1: *Our photo-art dataset: Photo-Art-50, containing 50 object categories. Each category is displayed with one art image and one photo image.*

5.2 A New Dataset and A Baseline

5.2.1 A New Dataset - *Photo-Art-50*

We provide a challenging, annotated image dataset for researchers to evaluate their cross-depiction techniques. This dataset contains 50 object categories, 90 to 138 images for each object with approximately half photos and half art images. These 50 objects all appear in Caltech-256 and a few also appear in PASCAL VOC Challenge [44] and ETHZ-Shape dataset [51], as shown in Fig 5-1. Part of the photo images are copied from Caltech-256, the rest are from Google search. Art images are searched by a few keywords to cover a wide gamut of depiction styles, *e.g.*, 'horse cartoon', 'horse drawing', 'horse painting', 'horse sketches', 'horse kid drawing', *etc.* Then we manually select images with a reasonable size of a meaningful object area. We further manually provide the ground-truth bounding boxes.

From the dataset we provided, it is not difficult to find there is a big difference in visual appearance exhibited by visual objects across depictive styles. This variation is typically much wider than for lighting and viewpoint variations usually considered for photographic images. Indeed, if we consider different ways to depict an object (or parts of an object) there is good reason to suppose that the distribution of corresponding features form distinct clusters.

To visualise how the photos and artworks are distributed, we display a few samples from horse and Eiffel Tower images in Fig 1-7. Differences between photo images and art images are significant, though human can easily recognise an object no matter how it appears in photo or in any kinds of art format. One may notice that the art domain exhibits larger diversity than photo images in the visual appearance. Such diversity is demonstrated with its larger variance in the feature space as shown in Fig 1-7.

K-L Divergence: In order to discover how much statistical difference exists between the feature distributions on the photo and art domains – and to make sure

Cross-domain datasets [62, 117]					*Photo-Art-50*
C-A	C-D	A-W	D-A	D-W	Photo-Art
0.079	0.271	0.239	0.292	0.047	**0.466**

Table 5.1: *Comparison of K-L divergence $\mathcal{D}(P_1, P_2)$ between domain pairs. Four domain sets in [117, 62]: C - Caltech-256, A - Amazon, W - WebCam, D - DSLR.*

our dataset is of value to the cross-depiction problem – we compute the symmetric Kullbeck-Liebler divergence between art and photo feature distributions.

We represent each image as a 5000-d BoW histogram with dense SIFT descriptors, which are then projected to a lower dimensional ($d = 10$) subspace using principle component analysis (PCA), and approximate the distributions using Gaussian densities $P_1 = \mathcal{N}(.|\mu_1, \Sigma_1)$ and $P_2 = \mathcal{N}(.|\mu_2, \Sigma_2)$, respectively. The K-L divergence of P_2 from P_1 is defined as

$$D_{\mathrm{KL}}(P_1||P_2) = \frac{1}{2}(tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T\Sigma_2^{-1}(\mu_2 - \mu_1) - d - ln(\frac{\det\Sigma_1}{\det\Sigma_2})). \quad (5.1)$$

We adopt the symmetrical K-L divergence, which is

$$\mathcal{D}(P_1, P_2) = (D_{\mathrm{KL}}(P_1||P_2) + D_{\mathrm{KL}}(P_2||P_1))/d. \quad (5.2)$$

A small $\mathcal{D}(P_1, P_2)$ means that the two distributions are similar.

Table 5.1 illustrates the K-L divergences between photo and art images in *Photo-Art-50*. To create a reference by which these divergences can be understood, we also compute the K-L divergences for domain pairs [62, 117] under different photographic conditions. The most similar domains are Caltech-Amazon and DSLR-Webcam, and the other three pairs are distributed very differently, which is consistent with the observation in [62]. However, even the largest K-L divergence in the cross-domain dataset is still not comparable with the photo-art divergence. This clearly tells that the photo-art distributions differ much more than distributions of photos capturing in different conditions.

5.2.2 Evaluation of Classification Baselines

We evaluate three baseline methods on our dataset. The first is bag-of-words (BoW), chosen because it is well known, widely used, and performs well on standard image classification problems. We assess the performance of this popular framework with different local descriptors, including the well-known self-similarity descriptor [121] designed for solving the cross-depiction matching problem.

The second baseline is the Fisher Vector(FV) [107] which extends BoW by going beyond count statistics. It has been shown to outperform BoW in most classification datasets.

Third, in order to investigate whether domain adaptation techniques help solve the cross-depiction problem, we evaluate two recent state-of-the-art domain adaptation techniques, called Geodesic Flow Kernel (GFK) [62] and Subspace Alignment (SA) [49].

Bag of Words

Using Bag of Words (BoW), each image is represented by the distribution of codeword occurrences.

Given a set of labelled training images, local descriptors are computed on a regular grid with multiple-sized regions. A codebook is constructed by vector quantisation of local descriptors with k-means clustering ($k = 1000$). Each image is first partitioned into L levels of increasingly fine cells ($L = 2$ in our experiments). A histogram of codeword occurrences is built for each cell. By concatenating all these histograms, each image is coded by a 5000 dimensional vector. A one-versus-all linear SVM classifier is then trained on a χ^2-homogeneous kernel map [148] of all training histograms. Given a test image the local features are extracted in the same way as in the training stage, mapped onto the codebook to build a multi-resolution histogram, which is then classified with the trained SVM. To explore the potential of different local features in the cross-depiction problem, we compare five types of local features.

- The popular ***SIFT*** [92] is a 128-dimensional vector created by stacking 8-bin orientation histograms on 4×4 cells weighted by an additional 2-D Gaussian function. We use the implementation of dense-SIFT in [147] and sample SIFT with four region sizes on a regular grid with 3 pixels step.

- ***Geometric Blur*** (GB) [12] describes local regions by geometrically blurring oriented edge maps. It is able to match object parts with very different appearance in two images, so we evaluate it on our cross-depiction dataset. Different from the densely sampling of SIFT, GB is extracted on regions centred on edge points. We randomly sample a few thousands edge points in 5 scales. The coefficients follow the original setup in [12].

- ***Self-similarity descriptors*** (SSD) [121] measure local self-similarity patterns by correlating a tiny local patch (typically 5×5) within a larger local region. It computes local correlations of patches rather than pixel values, and performance well at matching similar objects invariant to depictive styles. We include it in the BoW framework to see its behaviour in cross-depiction classification. We follow the default parameter settings from [24] except that we use 4 region sizes to capture a wider variation of local patterns. Instead of one single region size of radius 40, we extract SSIM with 4 radius sizes (28, 36, 44, 56) to capture wider variation of local patterns.

model		BoW					FV
train	test	SIFT	GB	SSD	HOG	edgeHOG	SIFT
Photo	Photo	83.69 ± 0.6	76.83±1.4	66.48±1.3	72.40±0.8	70.04±1.0	**87.42±0.5**
A+P	Photo	80.38 ± 1.1	71.94±1.1	57.85±0.9	64.67±1.4	63.25±1.3	**83.53±0.7**
Art	Photo	63.93 ± 1.1	59.90±0.8	38.89±1.6	42.45±1.1	50.13±1.4	**65.67±0.5**
Art	Art	74.25 ± 1.1	72.05±1.4	49.03±1.4	55.13±0.6	59.55±0.6	**76.74±0.5**
A+P	Art	69.47 ± 1.1	67.08±0.6	45.27±2.1	49.87±1.0	56.07±2.0	**72.82±1.0**
Photo	Art	43.78 ± 0.6	**50.42±1.4**	31.16±1.0	28.99±1.4	39.91±1.6	47.35±1.2

Table 5.2: *Comparison of categorisation performance on our proposed Photo-Art-50 dataset, with 30 images per category for training. Average correct rates are reported by running 5 rounds with random training-test split. 'A+P' stands for a mixture training set of 15 photo images and 15 art images.*

- The ***Histogram of Oriented Gradient*** (HOG) [34] is a vector of normalised histograms from tiled block regions. It is the most effective feature in the context of object detection and also the most favoured local feature in the context of sketch-based retrieval [90, 40, 41]. We compute HOG using the VLFeat [147] implementation. The gradients are quantised into 9 orientations and four cell sizes are used.

- We also include ***edgeHOG*** for comparison due to its effectiveness in sketch-based retrieval [71]. Unlike standard HOG which extracts the descriptor on the original image map, edgeHOG computes the gradient orientation histograms over edge maps. This helps improve matching performance between sketches and photo images.

We repeat the experiment 5 times, randomly selecting 30 images for training, using the rest for testing. Table 5.2 summarises the categorisation performances with BoW using different local features.

Discussion: Comparing different local descriptors, we can see that BoW with dense SIFT is the winner for all training-test combinations except 'Photo-Art' setting. Surprisingly, though SSD is designed for matching a common 'shape' regardless of their appearance, it performs poorly in classification on both same domain and different domains. Actually, SSD's inferior performance to HOG has also previously observed in the sketch-based classification task [72, 90]. EdgeHOG outperforms the standard HOG when art images are involved, which is consistent with the observation of [40, 71]. This may also explain the good performance of BoW-GB which also computes the descriptor on the edge map. When testing on the art domain, BoW-GB performs competitively and even outperforms BoW-SIFT when training on photo domain. This might result from the fact that edges posses some invariance across photo and artworks.

The same general trend appears across all descriptors; it can be explained by the degree of variation in the features, as evidenced in the KL-divergence Table 5.1. Training

on photos and testing on photos consistently returns the highest rates for all descriptors. Training on Art and testing on Art suffers some loss of performance, as expected. Yet what is most noticeable is all descriptors show a significant drop when trained on one depiction style and tested on another. This is evidence that BoW does not generalise well across depictive styles.

Fisher Vector

Fisher Vector (FV) is frequently used as a global image descriptor in visual classification. Instead of counting the codewords occurrence in BoW, it records the statistic information of local features inside each cluster.

Given a set of local feature vectors (we use SIFT) extracted from training images, let a K-component ($K = 256$ in our experiment) GMM fits the distribution of descriptors at first. The FV of an image is the stacking of the mean and covariance deviation vectors for each of the K clusters in the Gaussian mixture. We follow the improvement suggestions in [107] to apply the Hellinger's kernel to each dimension of the Fisher vector followed by l^2-normalisation. Like BoW, spatial pyramid is also applied in this experiment. The pyramid setting is the same as BoW. Then, a one-versus-all linear SVM classifier is trained on the Fisher vectors obtained from all training images.

Discussion: The performance of Fisher vector with SIFT is displayed in Table 5.2. Consistent with the observation in [107], it outperforms BoW-SIFT by 2-3% in all 'train-test' settings. In spite of such an improvement, FV still suffers from significant performance drop in the condition of different training and test depiction domains.

Due to the very different distribution of photo and art domains, it is natural to resort to the domain adaptation techniques. In the following section, we will investigate how well the domain adaptation could bridge the gap.

Domain Adaptive Benchmarks

In dealing with mismatched distributions between the training set and the test set, domain adaptive methods [63, 62, 49, 117, 61] have shown clear benefits. However, all these methods have been tested only on datasets containing photographs with different capture conditions. Intuitively, the distribution between photographs and artworks would have a greater variability. This intuition has been verified by the higher K-L divergency than standard cross-domain problem in Sec. 5.2. *It is unclear if the current domain adaptive methods can handle such diversity between photos and artworks.*

To find the answer, two state of the art methods are evaluated on our dataset:

- ***Geodesic Flow Kernel***GFK [62] models the source domain $\mathcal{S}$ and target domain $\mathcal{T}$ with lower dimensional linear subspaces and embeds them onto a Grassmann manifold $G(d, D)$. Let $P_{\mathcal{S}}, P_{\mathcal{T}} \in R^{D \times d}$ denote the basis of the PCA sub-

spaces for the two domains, respectively. The collection of all d-dimensional subspaces forms the Grassmann manifold $G(d, D)$. The geodesic flow is parameterized as a curve $\Phi(t), t \in [0, 1]$ between these two subspaces on the manifold, with $\Phi(0) = P_{\mathcal{S}}$ and $\Phi(1) = P_{\mathcal{T}}$. Thus this curve models the continuous deviation from the two domains. Different from [63], which only samples a number of intermediate subspaces, the original feature is projected into all these subspaces and concatenated into an infinite-dimensional feature vector: $z^{\infty} = \{\Phi(t)^T x : t \in [0, 1]\}$ which does not bias on either source or target domain. By using 'kernel trick', the similarity between two projected features is defined by their inner product as follows.

$$< z_i^{\infty}, z_j^{\infty} >= \int_0^1 (\Phi(t)^T x_i)^T (\Phi(t)^T x_j))dt = x_i^T \int_0^1 \Phi(t)^T \Phi(t) dt x_j = x_i^T G x_j \tag{5.3}$$

where $G \in R^{D \times D}$ is a positive semi-definite kernel matrix. The matrix G can be computed efficiently using singular value decomposition. Since the similarity or kernel is defined, we may classify images using either nearest neighbour or SVM classifier.

Similar to [62], the original features x_i are bag-of-words histograms. 5000-bin histograms are generated as in the previous section that used the SIFT descriptor, then normalised to have zero mean and unit standard deviation in each dimension. We project the original features onto 49 dimensional subspaces using PCA on each domain, i.e., $P_{\mathcal{S}}, P_{\mathcal{T}} \in R^{5000 \times 49}$. Follow the procedures of [62], we generate two variants of GFK kernels: GFK_PCA and GFK_LDA. **GFK_PCA** means that the original features are projected onto the 49 dimensional subspaces with PCA on each domain, i.e., $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$. In contrast, **GFK_LDA** replaces $P_{\mathcal{S}}$ with a supervised dimension reduction method - linear discriminant analysis (LDA) on source domain, and still PCA on the target domain. As LDA takes label information into account in the training stage, the source domain subspace possesses more discriminability for classification.

- ***Subspace Alignment*** (SA) [49] projects each source domain $\mathcal{S}$ and target domain $\mathcal{T}$ to its respective subspace X_s and X_t. Then, a linear transformation function is learned to align the source subspace coordinate system to the target one. To achieve this task, they use a *Subspace Alignment* approach. Basis vectors are aligned by using a transformation matrix M from X_s to X_T. M is learned by minimizing the following Bregman matrix divergence:

$$F(M) = ||X_S' X_S M - X_S' X_T||_F^2 = ||M - X_S' X_T||_F^2 \tag{5.4}$$

where $||\dot{|}||_F^2$ is the Frobenius norm. The optimal $M*$ is obtained as $M^* = X_S' X_T$

and this implies that the new target aligned source coordinate system is equivalent to $X_a = X_S X_S' X_T$. See Fernando *et al* [49] for mathematical details.

Other than the original features (**OrigFeat**), we also compare GFK and SA with another two no-domain-adaptation methods, the projected features with PCA bases from the source domain (**PCA_S**) and from the target domain (**PCA_T**), respectively. For the classifier, we implement both the simple First Nearest Neighbour (1-NN) and the more powerful SVM. In our experiment, the original features are 5000-bin histograms generated with BOW and SIFT descriptor as in the previous section.

Discussion: Fig. 5-2 compares domain adaptation methods with no-adaptation methods. As expected the SVM classifier consistently outperforms 1-NN. Features projected with PCA bases of target domain (PCA_T) always produce higher accuracies than those projected on the source domain (PCA_S), due to the better approximation of the distribution in the target domain. Using NN, the original feature yields the lowest accuracy and GFK_LDA the highest. However, the gain of GFK_LDA with either classifier is very little compared with PCA_T. Regarding SVM, the original feature surprisingly outperforms all the the other projected features, even the domain adaptive methods. The subspace alignment (SA) approach produces slightly lower results than GFK_PCA, no matter using NN or SVM. We also test higher dimensional projections, they yield slightly higher accuracies, but their performance rank remains the same.

Given these results, we conclude that state of the art domain adaption techniques (at least GFK [62] and SA [49]) show no improvement over PCA in the cross-depiction problem. As for BoW and DPM, this is likely due to the high dissimilarity between photo features and artworks features. Such dissimilarity has been measured with K-L divergence in Table 5.1. Since the main difference between artworks and photos originates in the local textures, it may cause the image presentations (histogram of SIFT words) to differ too much, This difference leads to either the case that no such smooth manifold exists or that the two subspaces are located too far apart on the manifold. Negative effects might occur with direct domain adaptation in such situations.

In the following section, we introduce a object class modelling way of using multi-labels to represent visual words and to capture the wide variations.

5.3 Models

Our model of a visual object class is based around a graph of nodes and edges. Like Felzenszwalb *et al* [46], we label nodes with descriptions of object parts, but we differ in two ways. Unlike them, we label parts with multiple attributes, to allow for cross-depiction variation. Second, we differ in using a graph that defines the spatial relationship between node pairs using edge labels, rather than a star-like structure in which nodes are attached to a root. Furthermore, we place weights on the graph which

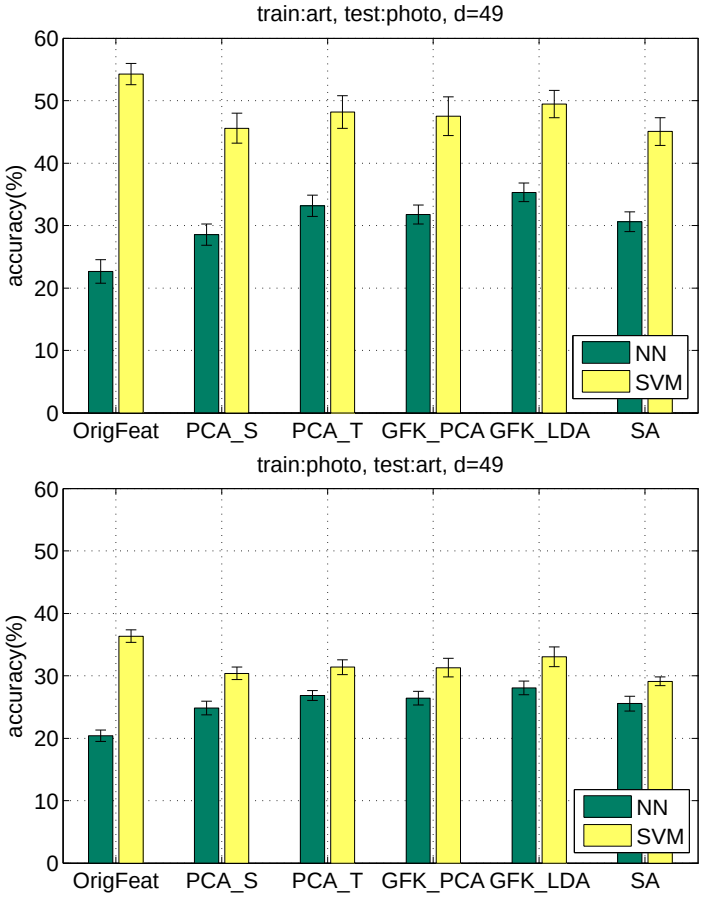


Figure 5-2: *Classification accuracies without (OrigFeat, PCA_S and PCA_T) and with (GFK_PCA, GFK_LDA, SA) domain adaptive methods on* Photo-Art-50. *Left: training on artworks, test on photographs. Right: training on photographs, test on artworks. The experiments are carried out with 30 images per class for training, repeated 5 times with random training-test split. 'OrigFeat' means classifying with the original 5000-bin BOW-SIFT histograms. Except OrigFeat, the rest methods are with 49 dimensional projected features.*

are automatically learned using a method due to [26]. These weights can be interpreted as encoding relative salience. Thus a weighted, multi-labeled graph describes objects as seen from a single viewpoint. To account for variation in points of view we follow [46, 65, 39] who advocate using distinct models for each pose. They refer to each such model as a *component*, a term we borrow in this paper and which should not be confused with the *part* of an object.

We solve the problem of inter-depictive variation by using *multi-labeled* nodes to describe objects parts. These multiple attributes are learned from different depictive styles of images, which are more effective than attempting to characterize all attributes in a monolithic model, since the variation of local feature is much wider than the changes usually considered for photographic images, such as lighting changes *etc.*

Moreover, it does not make sense that the parts of an object should be weighted equally during the matching for a part-based model. For example, for a person model,

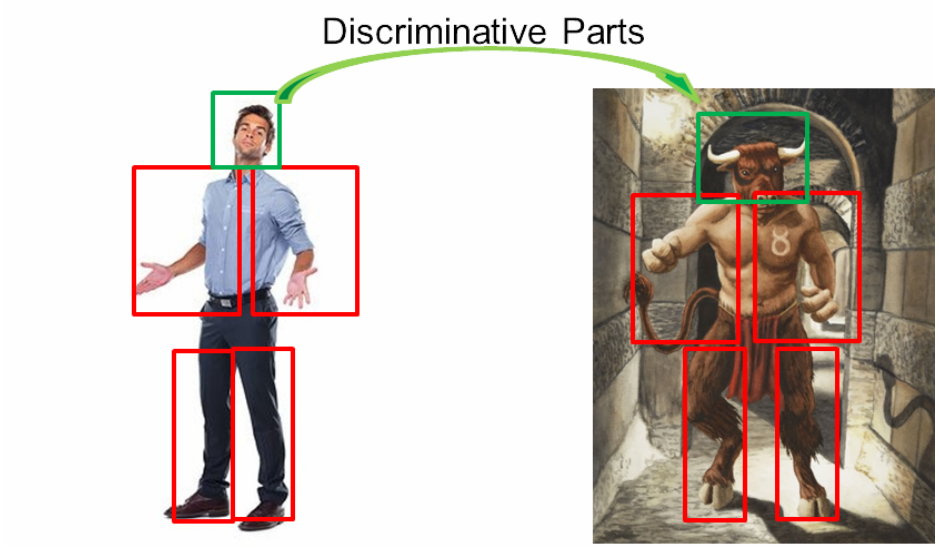


Figure 5-3: *Head is more discriminative than other parts in the matching - a person's arms are easily confused with a quadruped's forelimbs, but the head part's features are distinctive. In our model, parts are weighted according to its discriminatively.*

the head part should be weighted more than other parts like limbs and torso, because it is more discriminative than other parts in the matching - a person's arms are easily confused with a quadruped's forelimbs, but the head part's features are distinctive. Such an example is shown in Figure 6.2.2. Beside the discrimination of node appearance, the relative location, edges, should be also weighted according to its rigidity. For instance, the edges between the head and shoulder should be more rigid than the edges between two deformable arms. Hence, in our model, a weight vector β is learned automatically to encode the importance of node and edge similarity. We refer to it as the *discriminative weight* formulation for a part based model. This advantage will be demonstrated with evidence in the experimental section.

5.3.1 A Multi-labeled Weighted Graph Model

Our models are defined by a structural multi-labeled graph that approximately covers an entire object and nodes that cover smaller parts of the object.

A *multi-labeled graph* is defined as $G^* = (V^*, E^*, A^*, B^*)$, where V^* represents a set of nodes, E^* a set of edges, A^* a set of multi-labeled attributes of the nodes and B^* a set of attributes of edges. Specifically, $V^* = \{v_1^*, v_2^*, ..., v_n^*\}$, n is the number of nodes. $E^* = \{e_{12}^*, ..., e_{ij}^*, ..., e_{n(n-1)^*}\}$ is the set of edges. $A^* = \{A_1^*, A_2^*, ..., A_n^*\}$ with each $A_i^* = \{a_{i1}^*, a_{i2}^*, ..., a_{ic_i}^*\}$ consists of c_i attributes. It is easy to see that a standard graph G is a special case of our defined *multi-labeled graph*, which restricts $c_i = 1$.

A visual object class model $M =< G^*, \beta >$ for an object with n parts is formally defined by a multi-labeled model graph G^* with n nodes and $n \times (n-1)$ directed edges.

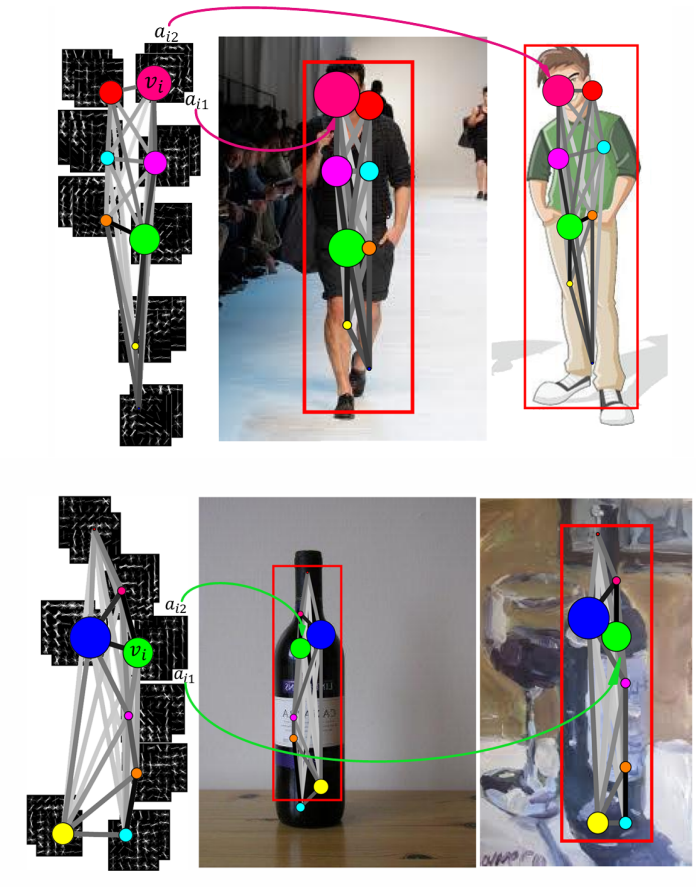


Figure 5-4: *Our multi-labeled graph model with learned discriminative weights, and detections for both photos and artworks. The model graph nodes are multi-labeled by attributes learned from different depiction styles (feature patches behind the nodes in the figure). The learned weight vector encodes the importance of the nodes and edges. In the figure, bigger circles represent stronger nodes, and darker lines denote stronger edges. And the same color of the nodes indicates the matched parts.*

And the weight vector $\beta \in \mathcal{R}^{n^2 \times 1}$ encodes the importance of nodes and edges of the G^*. And in this case, the A^*, a set of multi-labeled attributes of the nodes, describe the same part of the object but with a different instance, in practical terms, different depictive styles such as photos, paintings and cartoons. In other words, our model is a mixture model at the part level with a global graph structure of the arrangements of these parts. This bring us more robustness on the cross domain object detection and classification than normal part-based model. Both the model graph G^* and the weights vector β are learned from a set of labeled example graphs. Figure 5-4 shows two example models with their detections from different depictive style. The learning process depends on scoring and matching, so a description is deferred to Section 5.4.

We define a score function between a visual class model, G^*, and a putative object represented as a standard graph G, following [26]. The definition is such that the absence of the *VCM* in an image yields a very low score. Let Y be a binary assignment

matrix $Y \in \{0,1\}^{n \times n'}$ which indicates the nodes correspondence between two graphs, where n and n' denote the number of nodes in G^* and G, respectively. If $v_i^* \in V^*$ matches $v_a \in V$, then $Y_{i,a} = 1$, and $Y_{i,a} = 0$ otherwise. The scoring function is defined as the sum of nodes similarities (which indicate the local appearance) and the edges similarities (which indicate the spatial structure of the objects) between the visual object class and the putative object.

$$S(G^*, G, Y) = \sum_{Y_{i,a}\,=\,1} S_V(A_i^*, a_a) + \sum_{\substack{Y_{i,a}\,=\,1 \\ Y_{j,b}\,=\,1}} S_E(b_{ij}^*, b_{ab}), \tag{5.5}$$

where, because we use multi-labels on nodes we define

$$S_V(A_i^*, a_a) = \max_{p\,\in\,\{1,\,2,\,\ldots,\,c_i\}} S_A(a_{ip}^*, a_a), \tag{5.6}$$

with a_{ip}^*, the p_{th} attribute in $A_i^* = \{a_{i1}^*, a_{i2}^*, ..., a_{ip}^*, ...a_{ic_i}^*\}$, and S_A is the similarity measure between attributes.

To introduce the weight vector β into scoring, like [26], we parameterize Eq. 5.5 as follows. Let $\pi(i) = a$ denote an assignment of node v_i^* in G^* to node v_a in G, *i.e.* $Y_{i,a} = 1$. A joint feature map $\Phi(G^*, G, Y)$ is defined by aligning the relevant similarity values of Eq. 5.5 into a vectorial form as:

$$\Phi(G^*, G, Y) = [\cdots; S_V(A_i^*, a_{\pi(i)}); \cdots; S_E(b_{ij}^*, b_{\pi(i)\pi(j)}); \cdots]. \tag{5.7}$$

Then, by introducing weights on all elements of this feature map, we obtain a discriminative score function:

$$S(G^*, G, Y; \beta) = \beta \cdot \Phi(G^*, G, Y), \tag{5.8}$$

which is the score of a graph (extracted from the target image) with our proposed model $< G^*, \beta >$, under the assignment matrix Y.

5.3.2 Detection and Matching

To detect an instance of a visual class model (VCM) in an image we must find the standard graph in an image that best matches the given VCM. More exactly, we seek a subgraph of the graph G, constructed over a complete image, and is identified by the assignment matrix Y^+. We use an efficient approach to solve the problem of detection, which is stated as solving

$$Y^+ = \arg\max_{Y} S(G^*, G, Y; \beta), \tag{5.9a}$$

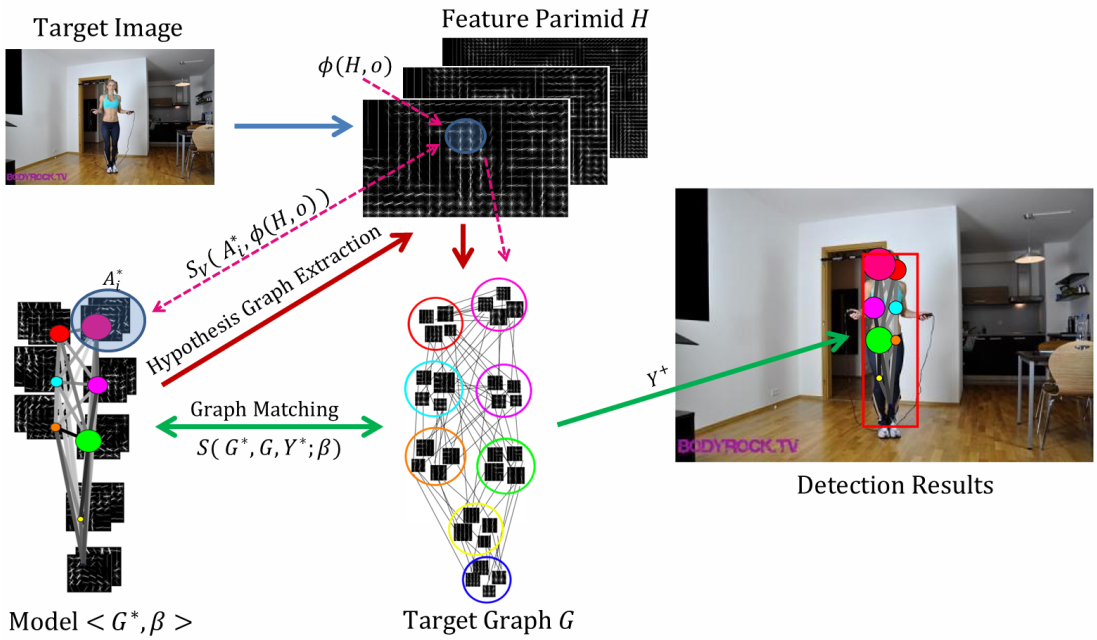


Figure 5-5: *Detection and matching process. A graph G will be firstly extracted from the target image based on input model $< G^*, \beta >$, then the matching process is formulated as a graph matching problem. The matched subgraph from G indicates the final detection results. $\phi(H, o)$ in the figure denotes the attributes obtained at position o.*

$$s.t. \left\{ Y \in \{0,1\}^{n \times n'}, \sum_{i=1}^{n} Y_{i,a} \leq 1, \sum_{a=1}^{n'} Y_{i,a} \leq 1 \right. \tag{5.9b}$$

where Eq.(5.9b) includes the matching constrains - only one node can match with at most one node in the other graph. To solve the NP-hard programming in Eq.5.9 efficiently, Torresani *et al.* [140] propose a decomposition approach for graph matching. The idea is to decompose the original problem into several simpler subproblems, for which a global maxima is efficiently computed. Combining the maxima from individual subproblems will then provide a maximum for the original problem. We make use of their general idea in an algorithm of our own design that efficiently locates graphs in images.

The graph G in Eq.(5.9a) is extracted from the target image and it varies with the input model graph G^*. For a target image, we first compute a dense multi-scale feature pyramid. The sampling scale in a feature pyramid is determined by a 3-tuple $(\lambda, \theta_{min}, \theta_{max})$, in which λ defining the number of levels of the pyramid, and $(\theta_{min}, \theta_{max})$ are used to define the maximal and minimal sampling scale, which are also relevant to the target image size S_{IM}, then we define $scale_{min} = \sqrt{S_{IM}/\theta_{max}}$, $scale_{max} = \sqrt{S_{IM}/\theta_{min}}$. These two values also imply the maximal and minimal part of the object in the target image we are able to detect. Then, the sampling scale at level $l \in \lambda$ can

be computed as

$$scale(l) = (l-1) \times [\frac{scale_{max} - scale_{min}}{\lambda - 1}] + scale_{min}. \tag{5.10}$$

In practical, we set $\lambda = 10, \theta_{min} = 10, \theta_{max} = 120$, which can cover a broad sampling scales range. Let H be a feature pyramid and $o = (x, y, l)$ specify a node position in the $l-th$ level of the pyramid. Let $\phi(H, o, scale(l))$ denote the vector obtained concatenating the feature vectors in the $scale(l)^2$ subwindow of H with top-left corner at n in row-major order. Below we write $\phi(H, o)$ since the subwindow scale are implicitly defined by the level l. Then, for each node v_i of the model graph G^*, we select kNN nodes from the feature pyramid H based on the similarity function $S_V(A_i^*, \phi(H, o))$ as the nodes of hypothesis graph G. These possible locations are used to create a graph of the image. The 'image graph' is fully connected; corresponding features from H label the nodes; spatial attributes label the edges. This creates graph G.

Having found G the next step is to find the optimal subgraph by solving Eq. 5.9. During this step, we constrain the node v_i^* of the model graph G^* to be assigned (via Y) only to one of the k nodes it was associated with. In our experiments, to balance the matching accuracy and computational efficiency, we set $k = 10$. The optimal assignment matrix Y^+ between the model $< G^*, \beta >$ and the graph G, computed through Eq. (5.9), returns a detected subgraph of G that indicates the parts of the detected object. A detection and matching process is illustrated in Fig 5-5.

5.3.3 Mixture Models

Our model also can be mixed using *components* as defined above and used in [46, 65, 39], so that different point of view (front/side) or poses (standing /sitting people) can be taken into account. A mixture model with m components is defined by a m-tuple, $\mathcal{M} = M_1, ..., M_m$, where $M_c =< G_c^*, \beta_c >$ is the multi-labeled VCM for the c-th component.

An object hypothesis for a mixture model specifies a mixture component, $1 \leq c \leq m$ and a matching result of model M_c. The score of this hypothesis is the score of the hypothesis graph G for the c-th model component. To detect objects using a mixture model we use the matching algorithm described above to find the best matched subgraph that yield high scoring hypothesis independently for each component.

5.4 Learning Models

Given images labeled with n interest points corresponding to n parts of the object, we consider learning a multi-labeled graph model G^* and weights β that together represent a visual class model. Because structure does not depend on fine-level details, we do not

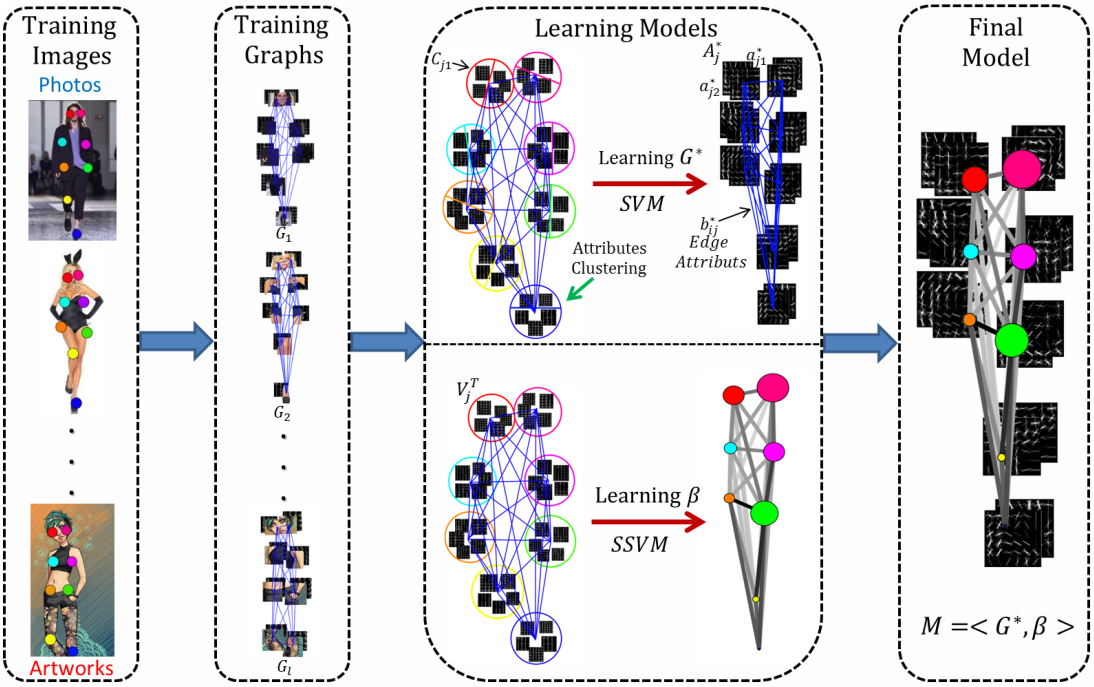


Figure 5-6: *Learning a class model, from left to right.(a): An input collection (different depictions) used for training. (b): Extract training graphs. (c): Learning models in two steps, one for G^*, one for β. (d): Combination as final class model*

(nor should we) train an ssvm using depiction-specific features. The model learning framework is shown in Figure 5-6 and the learning procedure is shown in 2.

5.4.1 Learning the Model Graph G^*

For the convenience of description, consider a class-specific reference graph $G^{\triangle}$ (note that a reference graph is not created but is a mathematical convenience only, see [26] for details) and a labeled training graph set $T = (< G_1, y_1 >, ..., < G_l, y_l >)$ obtained from the labeled images. In each $< G_i, y_i > \in T$, we have n nodes, $n \times (n-1)$ edges and their corresponding attributes, defined as $G_i = (V_i, E_i, A_i, B_i)$, and y_i is an assignment matrix that denotes the matching between the training graph and the reference graph $G^{\triangle}$. Then, a sequence of nodes which match the same reference node $v_j^{\triangle} \in G^{\triangle}$ are collected over all the graphs in T. We define these nodes as $V_j^T = \{v_{j,1}^T, v_{j,2}^T, ..., v_{j,l}^T\}$ in which $v_{j,i}$ means the j-th node in training graph G_i. Then, the corresponding attributes set A_j^T can be extracted from the corresponding G_i to be used to learn the model graph G^* via the following process.

To learn a node V_j^* in the model graph G^*, there are l positive training nodes V_j^T with their attributes A_j^T. All the attributes in A_j^T are labeled according to depictive styles. Instead of manually labelling the style for each image, we use K-means clustering based on chi-square distance to build c_j clusters automatically, C_{ji} denotes the i-th

cluster for A_j^T, and attributes in the same cluster indicate the similar depictive styles. Accordingly, the attributes A_j^* for the node $V_j^* \in G^*$ actually include c_j elements, $A_j^* = \{a_{j1}^*, a_{j2}^*, ..., a_{jc_j}^*\}$. For each a_{ji}^*, it is learned by minimizing the following objective function:

$$E(a_{ji}^*) = \frac{\lambda}{2}\|a_{ji}^*\|^2 + \frac{1}{N}\sum_{s=1}^{N}\max\{0, 1 - f(a_s) < a_{ji}^*, a_s >\} \tag{5.11}$$

from N example pairs $(a_s, f(a_s))$, $s = 1, ..., N$, where

$$f(a_s) = \begin{cases} 1 & \text{if } a_s \in C_{ji} \\ -1 & \text{if } a_s \in \mathcal{N}_j \end{cases} \tag{5.12}$$

where $\mathcal{N}_j$ is the negative sample sets for the node V_j^* and a_s is a node attributes from the training set. In our experiments, we use all the attributes that are in T but do not belong to A_j^T, and the background patch attributes to build the negative samples set. Hence, this learning process transfers to an SVM optimization problem, which is solved by using stochastic gradient descent [126]. Edges E^* and corresponding attributes B^* also can be learned in a similar way. We account for different depictive styles by constructing a distinct SVM for each one; so in effect the multi-labeled nodes in G^* are in fact multiple SVMs.

5.4.2 Learning the parameter β

The aim of this step is to learn a weight vector β to produce best matches of the reference graph $G^\triangle$ with the training examples $T = (< G_1, y_1 >, ..., < G_l, y_l >)$ of the class. Let $\hat{y}$ denote the optimal matching between the reference graph $G^\triangle$ and a training graph $G_i \in T$ given by

$$\hat{y}(G_i; G^\triangle, \beta) = \arg\max_{y \in Y(G_i)} S(G^\triangle, G_i, y; \beta), \tag{5.13}$$

where $Y(G_i) \in \{0, 1\}^{n \times n'}$ defines the set of possible assignment matrix for the input training graph G_i. Inspired by the max-margin framework [142] and following [26], we learn the parameter β by minimizing the following objective function:

$$L_T(G^\triangle, \beta) = r(G^\triangle, \beta) + \frac{C}{l}\sum_{i=1}^{l}\Delta(y_i, \hat{y}(G_i; G^\triangle), \beta). \tag{5.14}$$

In this objective function r is a regularization function, $\Delta(y, \hat{y})$ a loss function, drives the learning process by measuring the quality of a predicted matching $\hat{y}_i$ against its ground truth y_i. In our case, since all the nodes have been arranged with the same order of the reference graph, we have $y_i = I$, an identity matrix with size $n \times n$. The parameter C controls the relative importance of the loss term.

To optimize this function, we first separate the graph $G^{\triangle}$ from joint feature map $\Phi(G^{\triangle}, G, y)$. Since $G^{\triangle}$ is a single-labeled graph, from Eq. (5.7) we have

$$\Phi(G^{\triangle}, G, Y) = [\cdots; S_A(a_i^{\triangle}, a_{f(i)}); \cdots; S_E(b_{ij}^{\triangle}, b_{f(i)f(j)}); \cdots]. \tag{5.15}$$

We define similarity functions S_A and S_E are dot products of two attributes vectors:

$$S_A(a_i^{\triangle}, a_{f(i)}) = a_i^{\triangle} \cdot a_a,\ S_E(b_{ij}^{\triangle}, b_{f(i)f(j)}) = b_{ij}^{\triangle} \cdot b_{f(i)f(j)} \tag{5.16}$$

where $a_i^{\triangle}$ and $b_{ij}^{\triangle}$ correspond to the node and edge attributes of the reference graph respectively. Further, we define the attribute vector $\Theta(G^{\triangle})$ and $\Psi(G, y)$ as:

$$\begin{aligned} \Theta(G^{\triangle}) &= [\cdots; a_i^{\triangle}; \cdots; b_{ij}^{\triangle}; \cdots], \\ \Psi(G, y) &= [\cdots; a_{f(i)}; \cdots; b_{f(i)f(j)}; \cdots] \end{aligned} \tag{5.17}$$

where $\Theta(G^{\triangle})$ represents all the attributes of $G^{\triangle}$ and $\Psi(G, y)$ describes the corresponding attributes of G, according to the assignment y. This enables the attributes of $\Phi(G^{\triangle}, G, Y)$ to be factorized into $\Theta(G^{\triangle})$ and $\Psi(G, y)$, and then the score function can be rewritten as:

$$\begin{aligned} S(G^{\triangle}, G, y; \beta) &= \beta \cdot \Phi(G^{\triangle}, G, Y) \\ &= \beta \cdot (\Theta(G^{\triangle}) \odot \Psi(G, y)) \\ &= (\beta \odot \Theta(G^{\triangle})) \cdot \Psi(G, y) \end{aligned} \tag{5.18}$$

where $\odot$ denoted the element-wise product. By substituting $\mathbf{w} = \beta \odot \Theta(G^{\triangle})$ into Eq. (5.13), we obtain a linear form for the optimal assignment:

$$\hat{y}(G; \mathbf{w}) = \underset{y \in Y(G)}{\arg\max}\, \mathbf{w} \cdot \Psi(G, y), \tag{5.19}$$

This transforms the learning objective in Eq. (5.14) into a standard formulation of the structured support vector machine (SSVM):

$$L_T(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{l}\sum_{i=1}^{l} \Delta(y_i, \hat{y}(G_i; \mathbf{w}). \tag{5.20}$$

This function can be minimized by various optimization approaches to estimate the parameters $\mathbf{w}$ [142, 79, 133], which leads to the weight vector β.

Loss function. The loss function $\Delta(y, \hat{y})$ drives the learning process by measuring the quality of a predicted matching $\hat{y}$ against its ground truth y. We follow [26] to define

$$\Delta(y, \hat{y}) = 1 - \frac{1}{\|y\|_F^2} y \cdot \hat{y}, \tag{5.21}$$

Algorithm 2: Model Learning Procedure.

Input : Positive Examples $P = \{(I_1, B_1^1, B_1^2, ..., B_1^n), ..., (I_l, B_l^1, B_l^2, ..., B_l^n)\}$,
Negative Images $N = \{J_1, J_2, ..., J_m\}$
Output: Model $M =< G^*, \beta >$

run *Graph Extraction* on P to produce $T = (< G_1, y_1 >, ..., < G_l, y_l >)$
for $j := 1$ **to** n *(number of nodes)* **do**
  for $i := 1$ **to** l *(number of positive examples)* **do**
    extract features from part bounding box B_i^j
    Add to C_j
  end
  run $k - means(C_j, k)$, producing k clusters, denote as c_{jk}.
  for $s := 1$ **to** k **do**
    run Eq. 5.11 on the data of c_{jk} to learn a_{js}^* as the s^{th} label for j^{th} node in G^*.
  end
end
Output G^*.
learning β from T via. Eq. 5.14 and Eq. 5.20.
Output $M =< G^*, \beta >$

where $\| \cdot \|_F^2$ is the Frobenius norm.

Optimization. Many approaches have been proposed to train SSVM. This problem amounts to solving a convex quadratic program with an exponentially large number of constrains. Solutions for this optimization problem either: (i) reduce it to an equivalent polynomial-size reformulation and use methods like SMO [133] or general-purpose solvers: or (ii) work with the original problem by considering a subset of constrains, and employ cutting plane or stochastic sub-gradient methods. For solving the problem in Eq. (5.20), we use the efficient cutting plane method proposed by Joachims *et al.* [79]. This method differs from other SVM training approaches by considering individual data points as well as their linear combinations as potential support vectors. This leads to a smaller set of cutting plane models, and thus more efficient training.

5.4.3 Features

In general, any graph representation satisfying the condition of dot product similarity of Eq. 5.16, which leads to the linearization in Eq. 5.18, can be learned with the above learning approach. In this work, we use histogram distributions to represent the nodes and edges features in our graph model. The similarity value between two attributes in this graph is then computed as their dot product.

Node Attributes. For node attributes a, describing the local appearance of node v, we could adopt the histogram of gradient bins such as SIFT [92], HOG [34], and their variants, given their effectiveness. In our proposed model, we used a 31-dimension HOG

descriptor, following [46], which computes both directed and undirected gradients as well as a four dimensional texture-energy feature. The image patch is first divided into 6×6 non-overlapping cells. For each cell, a 1D histogram of gradient orientations over pixel in that cell is accumulated. Then, the gradient at each pixel is described into one of nine orientation bins, and each pixel votes for the orientation of its gradient, with a strength that depends on the gradient magnitude at that pixel. For colour image the channel with the largest gradient at that pixel is used. This leads to a *directed orientations histogram*. A second histogram of *undirected orientations* of half the size is obtained by folding the directed one into two. After that, both the directed and undirected histograms of each cell are normalized with respect to the gradient energy in a neighborhood around it. Let a block of the given HOG cell be a $2 * 2$ sub-array of cells, four normalisation factors are then obtained as the inverse of the norm of the four block that contain the cell. This leads to a $4 \times (2 + 1) \times 9$ dimension vector and then the resulting vector is projected down to $(2 + 1) \times 9$ elements by averaging corresponding histogram dimension. In addition, a four dimensional texture-energy feature is computed, for a total of $4 + 3 \times 9$ dimensional vector representing the local gradient information inside a cell.

Edge Attributes. We follow [26] to use the *histogram of log-polar bins* edge attribute to describe the geometric relationship between two nodes, as illustrated in figure 5-7.

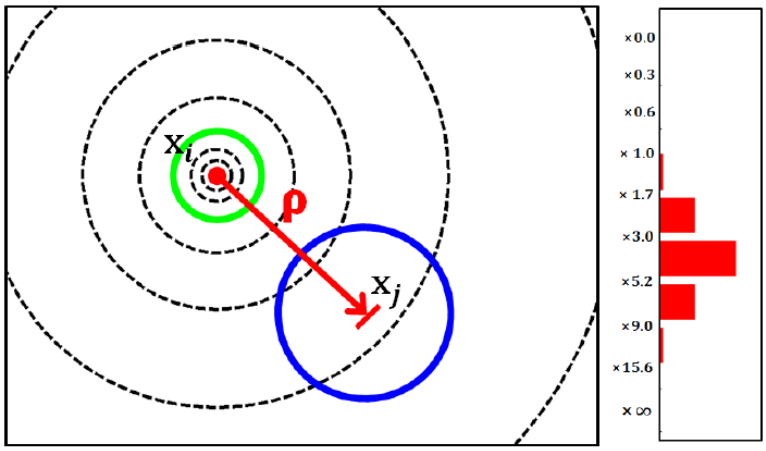
Consider an edge e_{ij} from node v_i to node v_j. The vector from v_i to v_j can be expressed in polar coordinates as (ρ_{ij}, θ_{ij}). Two histograms - one for length and another for angle - are build and concatenated to quantize the edge vectors. For length, we use uniform bins of size n_l in the log space with respect to the position of v_i, making the histogram more sensitive to the position of nearby points. The log-distance histogram L_{ij} is constructed on the bins by a discrete Gaussian histogram centred on the bins for ρ_{ij}:

$$L_{ij}(k) = t_L(k - m), \quad s.t. \;\; t_L(x) = \mathcal{N}(0, \sigma_L), \rho_{i,j} \in \text{bin}_\rho(m) \tag{5.22}$$

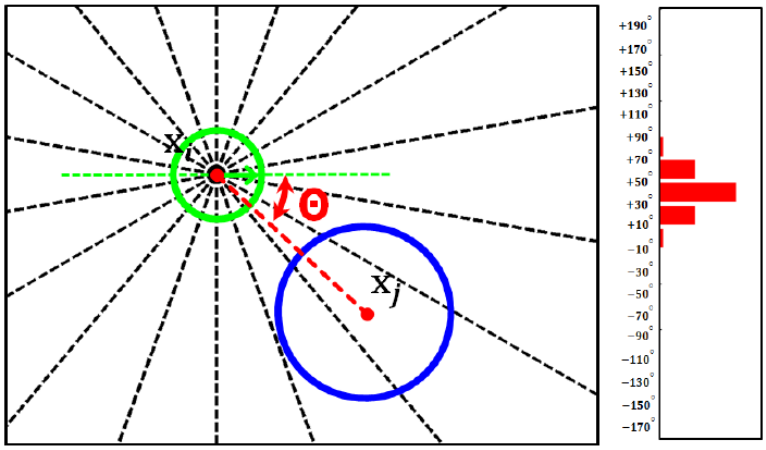
where $\mathcal{N}(0, \sigma_L)$ represents a discrete Gaussian window of size σ centred on μ, and $\text{bin}_\rho(k)$ denotes the kth log distance bin from the center of v_i. For angle, we use uniform bins of size $2\pi/n_P$. The polar-angle histogram P_{ij} is constructed on it in a similar way, except that a circular Gaussian histogram centered on the bin for $\theta_{i,j}$ is used:

$$P_{ij}(k) = t_P(k - m), \quad s.t. \;\; t_P(x) = \mathcal{N}(0, \sigma_P) + \mathcal{N}(\pm n_P, \sigma_P), \theta_{i,j} \in \text{bin}_\theta(m) \tag{5.23}$$

where additional Gaussian terms in $t_P(x)$ include the circular bins for angle. Then, the final histogram composed by concatenating the log-distance L_{ij} and the polar-angle P_{ij}, histograms is defined as the attributes for edges e_{ij}, so that $b_{ij} = [L_{ij}; P_{ij}]$, which is asymmetric ($b_{ij} \neq b_{ji}$). In this work, we used $n_L = 9, n_P = 18$ and $\sigma_L = \sigma_P = 5$.



(a) length of edge e_{ij} and its histogram attribute



(b) angle of edge e_{ij} and its histogram attribute

Figure 5-7: *Histogram of log-polar bins for edge attributes. Each histogram is represented by a discrete Gaussian window centered at a bin. (a) Log-distance ρ_{ij} (left) and its histogram with 9 bins (right). (b) Polar-angle θ_{ij} (left) and its histogram with 18 bins (right).[26]*

5.5 Experimental Evaluation

Our class model has the potential to be used in many applications, here we use detection and classification - and cross-depiction detection and classification in particular. All the experiments are evaluated on our Photo-Art-50 dataset.

5.5.1 Detection

In the detection task, we split the image set for each object class into two random partitions, 30 images for training (15 photos and 15 art) and the rest are used for testing. The dataset contains the groundtruth for each image in the form of bounding boxes around the objects. During the test, the goal is to predict the bounding boxes for a given object class in a target image (if any). The red bounding boxes in Fig. 1-11 are predicted in such way. In practice the system will output a set of bounding boxes with

corresponding scores, and we can threshold these scores at different points to obtain a precision-recall curve across all the test set. For a particular threshold the precision is the fraction of the reported bounding boxes that are correct detections, while recall is the fraction of the objects found. One scores a system by the average precision (AP) of its precision-recall curve across a test set. mAP(mean of the AP) is s the average AP over all objects.

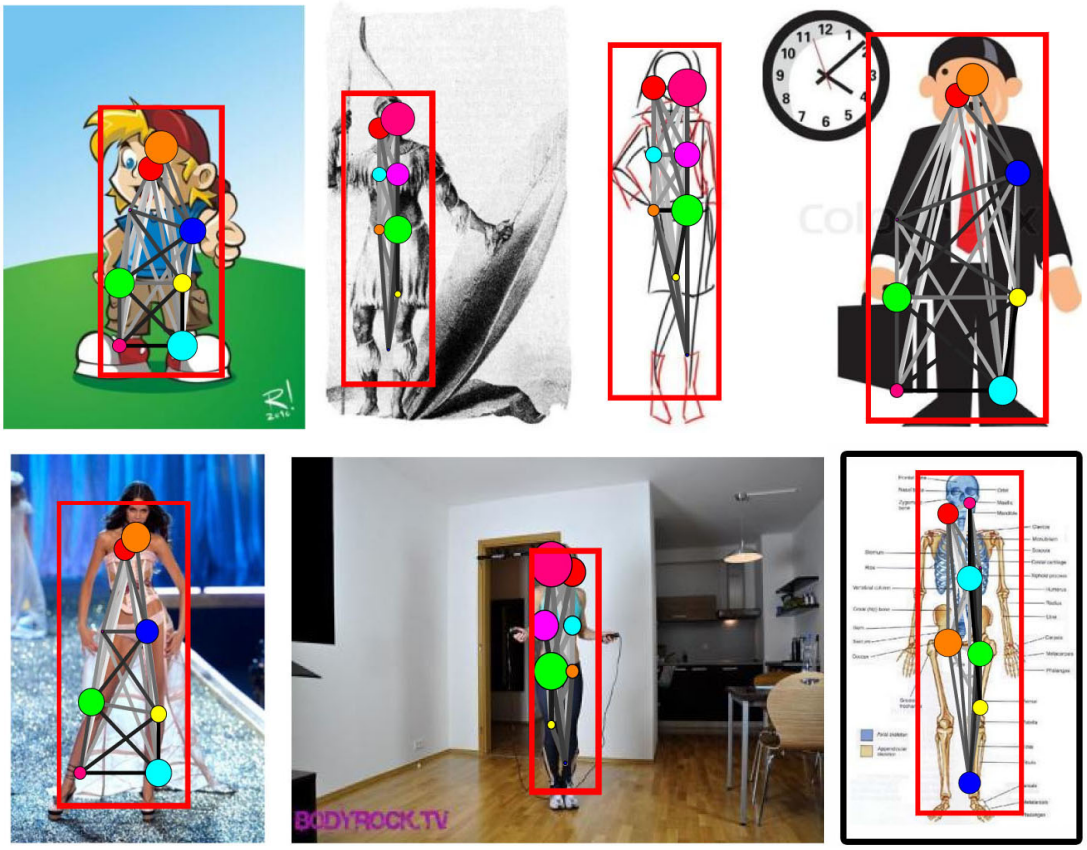
Since our learning process (in Sec. 5.4) needs pre-labeled training graphs, n distinctive key-points have to be identified in the target images. In our experiment, we set $n = 8$. In order to ease the labelling process, rather than using the manually labeling process, we instead use a pre-trained DPM model to locate the object parts across the training set, as only an approximate location of the labeled parts is enough to build our initial model. This idea is borrowed from [169], which uses a pictorial structure [118] to estimate 15 key-points for the further learning of a 2.5D human action graph for matching. Also notice that DPM is only used to ease the training data labelling process, it is not used in our proposed learning and matching process. During the test process, we match each learned object class model with the hypothesis graph extracted from an input test image, as detailed illustrated in Sec 5.3.2. However, we do admit that our success depend on the DPM performing well in the cross-depiction task when applying the DPM model directly on the training data (we need this step because we need part-location information in our training process). But if we can obtain ground-truth of the part locations of the training data, we do not need the DPM anymore and we believe our algorithm will be benefit from the more accurate training parts information.

The detection score is computed via Eq. (5.9) and the predicted bounding box is obtained by covering all the matched nodes.

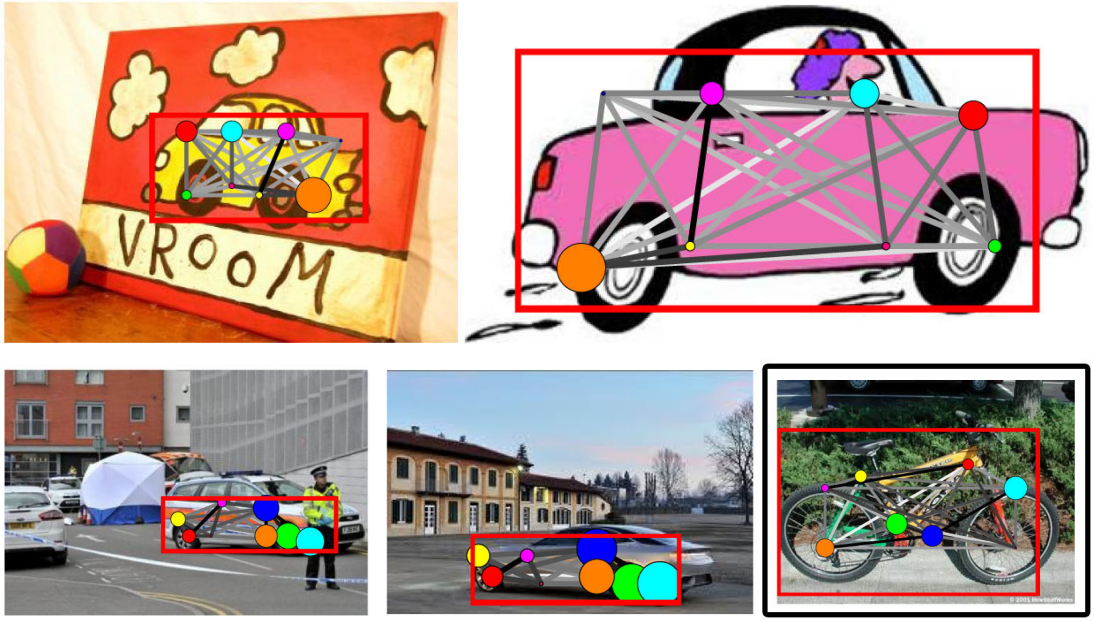
We trained a two component model, where the '*component*' is decided by the ground truth bounding box ratio as in DPM [46]. Each node in the model is multi-labeled by two labels (split automatically by *K-means* as illustrated in Sec. 5.4.1), that correspond to the attributes of the photo and art domains. Figure 5-8 shows some detections we obtain using our learned models. These results show that the proposed model can detect objects correctly across different depictive styles, including photos, oil paintings, children's drawings, stick-figures and cartoons. Moreover, the detected object parts are labeled by the graph nodes, and larger circles represent more important nodes, which are weighted more during the matching process, via β.

We evaluated different aspects of our system and compared them with a state-of-art method, DPM [46], which is a star-structured part-based model defined by a 'root' filter plus a set of parts filters. A two component DPM model is trained for each class following the setting of [46]. To evaluate the contribution of the mixture model and the importance of the weight β, we also implemented other two methods, multi-

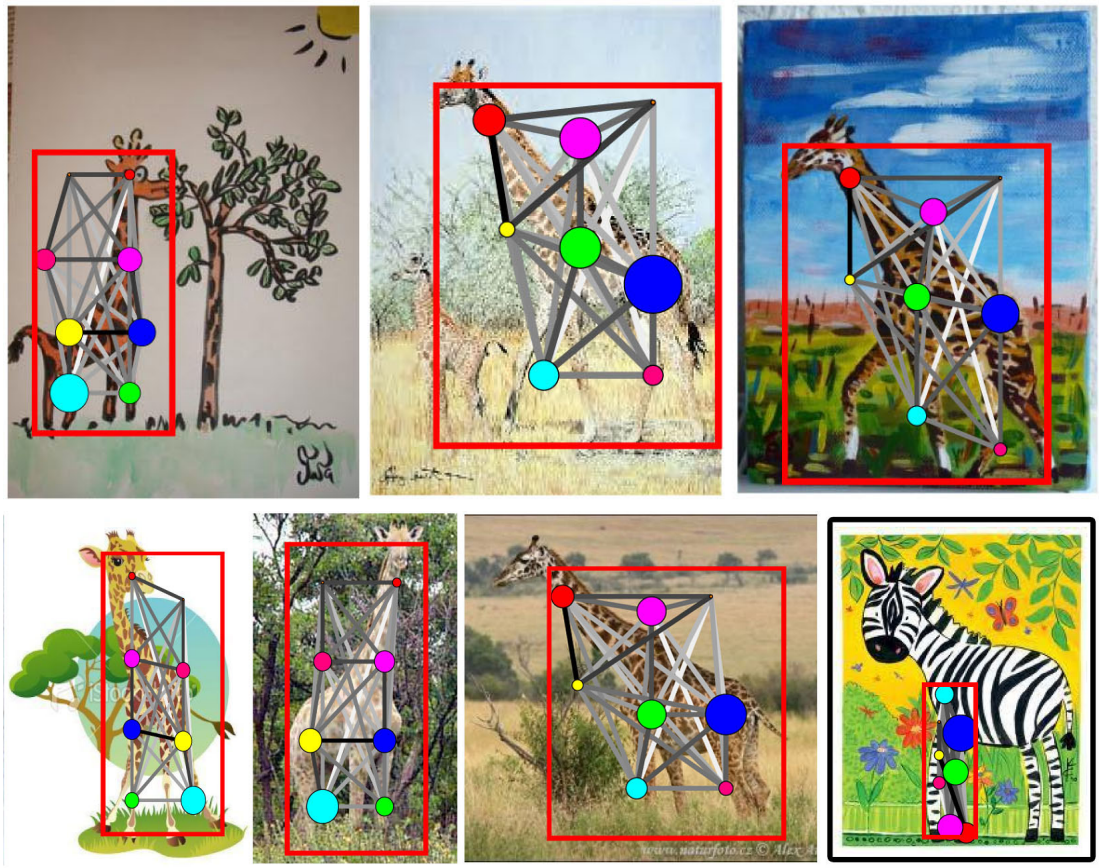
Person



Car



Giraffe



Bottle

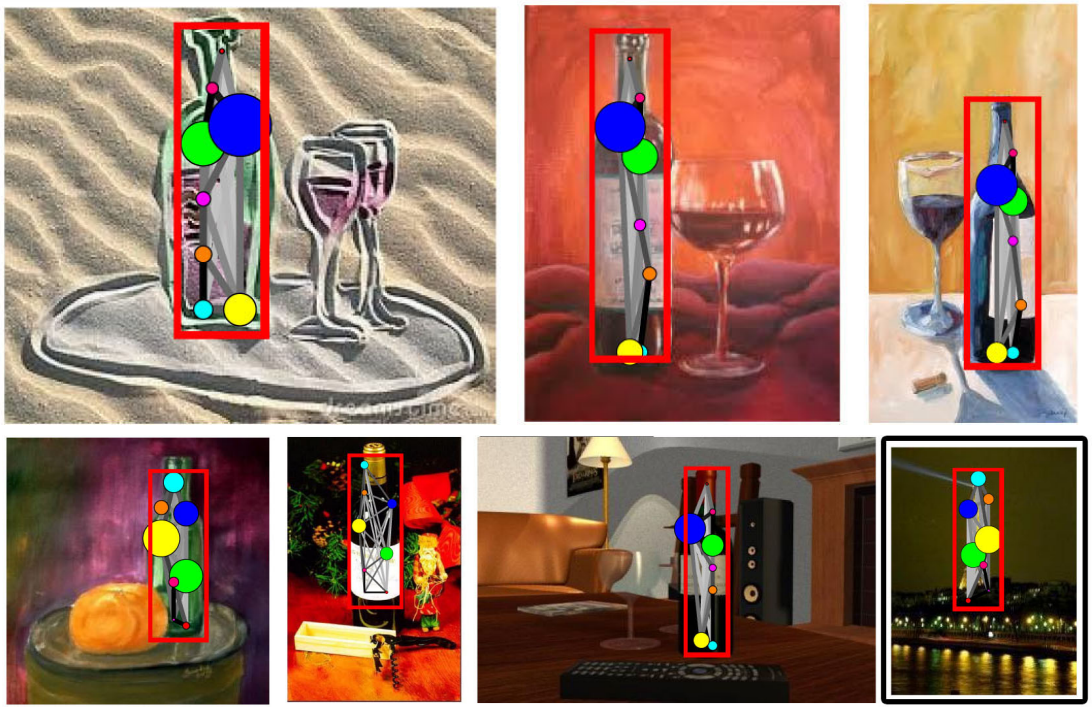


Figure 5-8: *Examples of high-scoring detections on our cross-depictive style dataset, selected from the top 20 highest scoring detections in each class. The framed images (last one in each class) illustrate false positives for each category. In each detected window, the object is matched with the learned model graph. In the matched graph, each node indicates a part of the object, and larger circles represent greater importance of a node, and darker lines denote stronger relationships.*

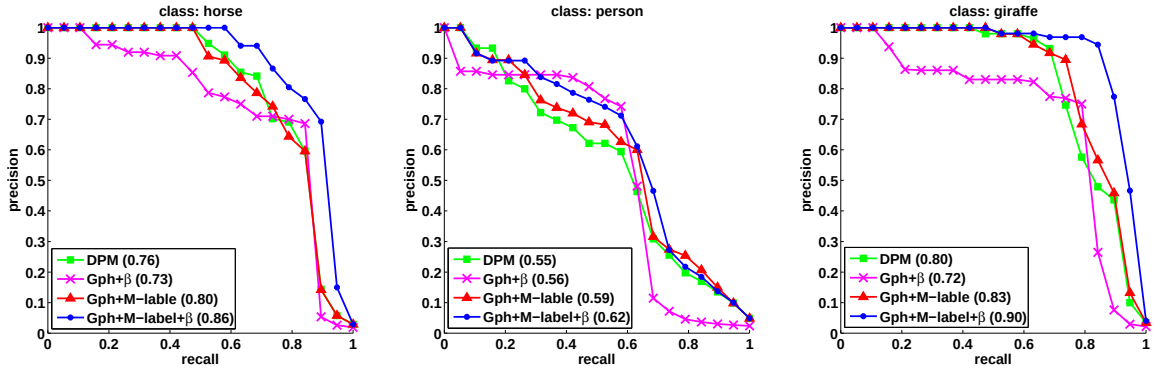


Figure 5-9: *Precision/Recall curves for models trained on the horse, person and giraffe categories of our cross-domain dataset. We show results for DPM, a single labeled graph model with learned β, our proposed multi-labeled model graph with and without learned β. In parenthesis we show the average precision score for each model.*

labeled graph without weight (Graph+M-label) and single-labeled graph with weight (Graph+β). The weight β can not be used on the DPM model, because it encodes no direct relation between nodes under the root.

Table 5.4 compares the detection results of using different models on our dataset. Our system achieves the best AP scores in 42 out of the 50 categories. Furthermore, our final mAP (.891) outperforms DPM (.835) by more than 5%. Figure 5-9 summarizes the results of different models applied on the person, horse and giraffe categories, chosen because these object classes appear commonly in many well-known detection datasets.

The PR-curve of other classes can be found in the Appendix C. We see that the use of our multi-labeled graph model can significantly improve the detect accuracy. Further improvements are obtained by using discriminative weights β.

5.5.2 Classification

Our proposed model can also be adapted for classification. Like any classification task, ours consists of two main steps, training and testing. Training requires of learning a class model, exactly the same procedure as in the previous section. The testing process determines the class by choosing the class which has the best matching score with the query image.

Using our dataset we conduct experiments designed to test how well our proposed class model generalised across depictive styles. Like the detection experiments, we randomly split the image set for each object class into two partitions, 30 images for training (15 photos and 15 artworks) and the rest are used for testing. Unlike from the detection task, we test on photos and artworks separately to compare the performance on these two domains. The classification accuracy is determined as the average over 5 random splits.

Methods	Art	Photos
BoW[147]	69.47 ± 1.1	80.38 ± 1.1
DPM[46]	80.29 ± 0.9	85.22 ± 0.6
Our	$\mathbf{89.06 \pm 1.2}$	$\mathbf{90.29 \pm 1.3}$

Table 5.3: *Comparison of classification results for different test cases and methods.*

For comparison with alternative visual class models we compare with two other methods: BoW and DPM. BoW classifier is chosen because it performs well and will help us assess the performance of such a popular approach to the problem of cross-depiction classification. We follow Vedaldi *et al* [147] using dense-sift features [15] and K-means ($K = 1000$) for visual word dictionary construction. Finally, it uses a SVM for classification. The second is the DPM [46], adapted to classification. Given a test image, the object with highest score among 50 class models is the output class label.

Classification accuracy of different methods in various testing cases, are shown in table 5.3. It shows that our method outperforms the BoW and DPM method in all cases, especially when the test set are artworks only. Our multi-labeled modelling method effectively train nodes of the graph in separately depictive styles and then combine them in a mixture model to global optimization. Experimental results clearly indicate that our mixture model outperforms state of the art methods which attempt to characterize all depiction styles in a monolithic model. We also made tests on some of the cross-domain literature we cited such as [121, 164] and a method that is not depend on photometric appearance, using the edgelets [51]. But none of them work well on such a high-variety depiction dataset. We report DPM and BoW (with Dense-SIFT) only because they consistently out-perform those methods, with no additional input from us.

Our system is implemented by matlab, running on a Core i7 CPU@2.67GHz×8 machine. The average training time for a single class is 4 to 5 minutes (parts labelling process is not included). The average testing time of a single image is 4.5 to 5 minutes, since the graph matching takes long time.

5.6 Discussion and Conclusion

We make three conclusions for this chapter, one in line with prior art, the other we have not seen elsewhere. The "in-line" conclusion is that structural information is important, even in a single domain. This is not new, but our experiment on classification provides further evidence that a graph model which captures inter-nodes relations directly is of value: the star-like graph of DPM out-performs BoW in classifying photographs alone, and we out-perform DPM in the same task. More importantly, our paper provides evidence that multi-label nodes are useful representations in coping with features that

exhibit very wide, possibly discontinuous distributions. There is no reason to believe that such distributions are confined to the problem of local feature representation in art and photographs; it could be an issue in many cross-domain cases. Our final conclusion is then: computer vision is likely to benefit both theoretically and in applications, if the question of recognition regardless of depiction is more fully understood.

Our models are already capable of representing highly variable object classes depicted in wide styles by using a deeper part hierarchies, but we would like to move towards richer models. In the future, we would like to build a model whose parts or nodes are reusable among different object models. A new model can be assembled from several pre-trained models, then used for detecting and classifying the new object. This will be discussed in the future work.

	us-flag	bat	beer-mug	boom-box	butterfly	camel	wagon	crab
DPM[46]	.871	.338	**.956**	.859	.718	.797	.897	.640
G+β	.743	.343	.729	.683	.375	.420	.662	.633
G+ml	.913	.423	.947	.913	.757	.816	.914	.723
G+ml+β	**.917**	**.456**	.954	**.929**	**.839**	**.881**	**.942**	**.741**

	globe	eiffel-tower	elephant	fried-egg	frying-pan	giraffe	goldfish	hamburger	head-phones
DPM[46]	.961	.952	.853	.618	.757	.803	.901	.809	.808
G+β	.846	.798	.538	.627	.733	.716	.556	.650	.669
G+ml	.932	.971	.875	.778	.812	.831	.913	.883	.853
G+ml+β	**.993**	**.981**	**.887**	**.851**	**.838**	**.907**	**.962**	**.908**	**.879**

	horse	balloon	hourglass	skeleton	ice-cream	ketch	laptop	lightbulb
DPM[46]	.764	**.955**	.925	**.985**	.816	.905	.926	.751
G+β	.733	.755	.881	.926	**.930**	.718	.780	.705
G+ml	.799	.899	.953	.956	.894	.929	.954	.725
G+ml+β	**.860**	.930	**.956**	.968	.911	**.976**	**.964**	**.807**

	mandolin	menorah	bike	palm-tree	penguin	person	pyramid	refrigerator	rotary-phone
DPM[46]	.385	.882	.975	.899	.735	.554	**.840**	.731	.900
G+β	.487	.879	.933	.794	.599	.555	.719	.561	.713
G+ml	.415	.901	.996	.921	.742	.587	.760	.788	.916
G+ml+β	**.491**	**.933**	**.997**	**.936**	**.825**	**.616**	.808	**.820**	**.928**

	starfish	sunflower	superman	swan	teapot	teddy	teepee	tower-pisa
DPM[46]	.922	.881	.720	**.923**	.968	.962	.807	.964
G+β	.713	.766	.584	.723	.632	.814	.872	.849
G+ml	.942	.903	.758	.859	.992	.985	.922	.981
G+ml+β	**.965**	**.923**	**.791**	.915	**.993**	**.991**	**.944**	**.991**

	umbrella	wash-machine	watch	windmill	bottle	zebra	car	face	**mAP**
DPM[46]	**.924**	.957	.736	.889	.892	.914	.948	**.906**	.835
G+β	.724	.974	.744	.739	.639	.751	.851	.790	.711
G+ml	.794	.982	.774	.899	.911	.961	.965	.886	.858
G+ml+β	.898	**.985**	**.794**	**.965**	**.956**	**.973**	**.974**	.899	**.891**

Table 5.4: *Detection results on our cross-depictive style dataset (50 classes in total): average precision scores for each class of different methods, DPM, a single labeled graph model with learned β, our proposed multi-labeled model graph with and without learned β. The mAP (mean of average precision) is shown in the last column.*

CHAPTER 6

CONCLUSIONS

Cross-depiction object recognition is significantly under-explored by the Computer Vision community. Yet there is a deep appeal in not discriminating between depictive styles, not just because it echoes an impressive human ability but also because it opens new applications. Providing machines with the ability to deal with objects regardless of the way in which they are depicted forces us to consider representations of objects that are more general than appearance in any one depictive style (including photography).

In this thesis we examined models for recognising objects across different depictive styles. Specifically, we tested the hypothesis that

> ***object class representation is the key to solve the cross-depiction object recognition problem.***

Evidence supporting this hypothesis is found in the significant accuracy drop exhibited by state-of-art – especially when training and testing across different depictions (see Chapter 4 and Chapter 5). However, our representation exhibits no such fall, remaining stable across depictions.

The drop of state-of-art is because Computer Vision is premised on (photographic) appearance. For example, the formation of visual codewords in Bag-of-Words assumes low variation in feature appearance and is biased towards 'photographic words'. However, appearance exhibits a much wider variation in the cross depiction problem and evidence is given in Chapter 1 - we show the inter-depiction divergencies are bigger than inter-category divergencies.

Humans are able to recognise objects in a seemingly unlimited variety of depictions. For example, the stick-man in Figure 1-3 and the breakfast-face shown in Figure 1-10. Both these examples suggest that the topology is important for recognition. We use graphs to represent topological structures. In Chapter 4, we use a median graph model, and a weighted graph is proposed in Chapter 5.

However, structure alone only defines very wide classes [167]. Hence, other information is needed for finer-grade classification. Shape is a natural representational element. We showed that an object class can be characterised by the qualitative shape of object parts and their structural arrangement. However, this model is still not strong enough when more depictive styles are included. To account for the wider variation in visual appearance distributions, we use multi-labeled graphs.

The above representation is stable for most rigid objects. But we acknowledge that these representational elements may not work with some natural objects such as water, smoke etc. Then there must be some other ways to represent these kind of objects.

In the following sections, we summarized the technical works followed by a future plan and final listing of the conclusions of this thesis.

6.1 Summary of Work

With the challenges raised in Chapter 1, we first reviewed the state of the art in object recognition in Chapter 2, showing that although many algorithms have been proposed to address the object detection and classification, there is very little research in computer vision on the problem of recognising objects regardless of depictive style. In this thesis, we make an effort to fill this literature gap.

Our first attempt motivated by the artistic methodology and psychology. We employed pattern recognition technology in Chapter 3 to find out whether common simple shapes exist in natural images. The discovery of the work is unique, so far as we know: regions in image segmentation naturally form classes that correspond to simple, easily recognised shapes, upon given appropriate region descriptions and well-designed classifiers. We employed two shape descriptors and three different shape producers(segmentation methods). And mean-shift and a self-designed clustering algorithm are used to classify regions into primitive shapes. In short, we have provided empirical evidence to suggest that *some of regions in segmentations can be classified as primitive shapes, upon given appropriate region descriptions and well-designed classifiers.*. They are 'features in the signal', and as such can be of use to many applications in computer vision. An application of scene classification is implemented based on this research.

We argue that an object class can be characterised by the qualitative shape of object parts and their structural arrangements in Chapter 4. Hence we used a graph of nodes and arcs in which qualitative shapes such as triangle, square, and circle to label the nodes. More exactly our model is a hierarchy of levels, yielding a coarse-to-fine representation. Each level contains an undirected graph of nodes and arcs. Nodes between levels are connected via parent-child arcs, which are directed. Child nodes are nested inside their parent. We took the lead in showing that *it is possible to learn models*

of object classes that generalise across depictive styles, in the sense that it is possible to learn a model using one style but classify objects depicted in other styles. Experiments we presented in Chapter 4 show that our proposal method performs better than the traditional visual appearance based method in cross-depiction problems (including to unseen depictive styles), in mixed problems, and in art-only problems.

With a much more challenging cross-depiction dataset was established in Chapter 5. It forced us to explore ways to capture the wide variation in visual appearance exhibited by visual objects across depictive styles. In this chapter, we first showed the gap between photorealist images and artworks exists both visually and statistically. Based on this new dataset, we evaluated leading recognition and detection techniques and two state-of-the-art domain adaptive methods for cross-depiction tasks; no one performs well. We then introduced a weighted multi-labeled graph model in which we account for the wide variation in feature distribution and experiments show that our representation is able to improve upon Deformable Part Models for detection and Bag of Words models for classification.

After several experiments, we draw two conclusions, one is consistent with prior art, the other we have not seen elsewhere. The "in line" conclusion is that structural information is important, even in a single domain. This is not new, but our experiment on classification provides further evidence that a graph model which captures inter-nodes relations directly is of value: the star-like graph of DPM out-performs BoW in classifying photographs alone, and we out-perform DPM in the same task. More importantly, our paper provides evidence that *multi-label nodes are useful representations in coping with features that exhibit very wide, possibly discontinuous distributions.* There is no reason to believe that such distributions are confined to the problem of local feature representation in art and photographs; it could be an issue in many cross-domain cases.

6.2 Future Work

Previous chapters have shown that we made our efforts to explore and study this new, challenge and important area in several aspects. We try to open a new area but not to close one. Our research and results are a first step towards depiction invariant modelling. There are several works need to be done in the future. For example, we need to more fully investigate the way in which the distribution of the description of a single object part is represented. Currently we use multi-label nodes, which is a discrete and frequentist approach to what is a potentially continuous and Bayesian problem; whether the additional computational costs of a more principled approach is of value is an open question, and one that can only be properly investigated with a much larger database than is available to anyone at present.

Moreover, the number of nodes in our current graphical model is fixed to 8, to

balance the accuracy and efficiency. We plan to vary the number of nodes for each model on a per-class basis. Some simple objects like bottles, cups may need less parts to represent, while some complicated objects such as person, cars and bicycles needs more parts. We suspect this may be related to the changing rate of object boundary, but more experiments and observation are needed.

Deformable part-based model [46] is an elegant framework to model visual object class. Its effectiveness in cross-depiction object detection and classification has been shown in table 5.3 and 5.4. Although it performs not as good as ours, we believe the main reason is caused by the single labelled part filters. Hence, if we can join our multi-labeled theory with the DPM, ie, multi part filters at each part location in each DPM model, we believe the results will be boosted, not to mention that the latent SVM optimization is a much more faster process than graph matching. Most recently, we found Yang et al proposed a DPM model allow for flexible mixtures-of-parts [168], which can be used as a good start to implement multi-labels DPM.

Primitive shapes can also be included in our current multi-labelled model. In this way, both the abstraction information of the region and the fine detailed local features are included in our model. Based on the conclusion we made in Chapter 4 – abstraction brings more robustness to non-salient variations, the improved model may improve the performance on cross-depiction object detection and classification.

And some other interesting directions for the future would be in applications, we list some showing as following sub-sections.

6.2.1 Incremental learning of Models

In the current work, we are applying an off-line training strategy, which means the object class model is learned from fixed number of training images once. However, as we know, there seems no bounding of depictive styles, so we want an visual class model can be updated when new data come in. In other words, we want an incrementally way to learn the visual class models.

To fulfill this task, at least three challenges have to be addressed. At first, updating the appearance model. The part appearance templates in our current model are learned using a leaner SVM. So if we want to learn the appearance model incrementally, we need an incrementally SVM. Current state-of-the-art of incremental support vector machine learning is proposed by Cauwenberghs and Poggio [21]. They consider incremental learning as an exact on-line method to construct the solution recursively, one point at a time. The key is to retain the Kuhn-Tucker (KT) conditions on all previously seen data, while "adiabatically" adding a new data point to the solution. The second challenge is to update the structure. Since the structural information of our model is learned using SSVM, we need an incrementally SSVM. To our knowledge, incrementally SSVM is still a blank in computer vision and machine learning. The third problem

need to be addressed for incremental learning models is how to decide the styles of the incrementally input images. The essence of our model is 'multi-labels' - we learn the object part templates separately according to the depicted styles. When a new data comes in, we need to decide which style it belongs to or it might be a new style we have never seen before in the model. Hence, a style classifier is required and it also need to has the ability to detect new styles.

6.2.2 Assembly Modelling and Generalised Matching

Assembly modelling is a technology used by CAD (computer-aided design) system to handle multiple parts within a product. The parts within an assembly are represented as solid models. Wouldn't it be useful if a new visual class model can be assembled by the parts of other pre-trained models? In other words, parts are reusable among different object models.

Our models are already capable of representing highly variable object classes depicted in wide styles by using a deeper part hierarchies, but we would like to move towards richer models. In the future, we would like to build a model whose parts or nodes are reusable among different object models. A new model can be assembled from several pre-trained models, then used for detecting and classifying the new object. We call this as "generalised matching", by which means we can detect the presence of new objects we have not trained on.

This application will be useful when one wants to detect some 'unusual' objects, such as a 'people with horse mask' or a 'people with bull head' as we shown in the Figure . The number of this kind of objects is limited, so it is hard to find enough such objects to train a model. Then, assembly modeling will be helpful - such a model can be assembled by a 'people body part' model and a 'horse head part' model. It will also be very helpful when people want to detect and recognise the visual objects not existed in the real world, Mythological creatures (Figure 1-4), for example, have never existed but are recognisable nonetheless. Most of these objects models can be assembled by several other real existing objects' parts. These objects are the target of generalised matching.

Moreover, this research will extend our work from the challenges of 'Category Level' to the 'Semantic Level', ie, to address the ambiguous meanings of the same image. There are mainly two reasons to cause the ambiguous. The first one is because of the occlusion. Take the rabit/duck head in Figure 2-3 as an example, if the body part (either a rabbit body or a duck body)is also shown, there will be no ambiguous anymore. This is also true for the face/candle example. Another reason that cause the ambiguous is the multiple high response between the query image and trained models. This actually happens in both human and computer vision. For human, when we observe an ambiguous image, there must be multiple models in our memory have been

recalled. For a computer, there must be multiple pre-trained models which are matched with the query image with very close and high recognise score. For example, the input of duck/rabit head can at least match a duck and a rabbit model with high response. A normal way we deal with this problem in computer vision is simply to choose the highest one or set a threshold to avoid the ambiguous. However, it is worthy to think about how to address this in an alternative way. Our generalised model provides an option since we allow for multiple responses for a query.

6.2.3 Convolutional Neural Networks for Cross Depiction Object Modelling

Deep convolutional networks have a long history in computer vision. More recently, these networks have achieved competition-winning numbers on large benchmark datasets consisting of more than one million images. Deep convolutional activation features (DeCAF) [77] is a new visual feature defined by convolutional network weights learned on a set of pre-defined object recognition tasks. It is also famous because of its domain adaptation performance so it is good to see how deep feature performs on our such tasks.

Some researches have been done in this area actually, for example, in [32], Crowley and Zisserman show that object classifiers, learnt using Convolutional Neural Networks (CNNs) features computed from virous natural images sources, can retrieve paintings containing these objects with great success. Specifically, a CNN network, which consists of 5 convolutional layers and 3 fully-connected layers, is trained solely using ILSVRC-2012 (Large Scale Visual Recognition Challenge). A feature vector of an image is obtained by passing it through the network and then the output of the penultimate layer is recorded. Then, linear-SVM classifiers are learnt using linear-SVM training data per class in a one-vs-the-rest manner.

We use the exactly same configuration in [32] to test on our cross-depiction dataset, producing results in table 6.1.

From the results, it clearly shows that deep features behave incredibly well in our dataset and it shows great potential that the cross-depiction problem can be addressed via CNNs-based framework. We are planning to build a bigger and more challenge cross-depiction dataset which allow us to run more test. And may be a fine-tuning process can be added as Girshick et al did in [60].

6.2.4 Artistic Theme Understanding

There is always a reason (may be not single reason) why artist draw some things or scenes. Artists want to send a message to the viewers through paintings or art works. Some artists may want to present their own mood in the painting, some others may just want to represent a historical events. And many art works are created because

model		BoW	FV	DPM	Our	CNNs-fc6[32]
train	test	SIFT	SIFT	-	-	-
Photo	Photo	83.69	87.42	87.78	-	**96.95**
A+P	Photo	80.38	83.53	85.22	90.29	**96.23**
Art	Photo	63.93	65.67	77.59	-	**90.50**
Art	Art	74.25	76.74	83.02	-	**89.24**
A+P	Art	69.47	72.82	80.29	**89.06**	87.13
Photo	Art	43.78	47.35	68.30	-	**72.54**

Table 6.1: *Comparison of categorisation performance on our proposed Photo-Art-50 dataset, with 30 images per category for training. 'A+P' stands for a mixture training set of 15 photo images and 15 art images. The BoW and FV results are from section 5.2.2. DPM and our results from section 5.5. Please note that we didn't evaluate our method on 'single domain training' case, such as Photo-Photo, Art-Photo, Art-Art and Photo-Art because our method is designed to have multiple depictive styles input. CNNs-fc6 means the features are from the fully-connect layer 6 in the AlexNet, more details can be found in [32].*

of revisionary, political or social issues. Not to mention that there are some paintings are about nature or human experience. And of course some works are just for visual delight.

Human have the ability to crack and receive the message from art works. Although one thousand people may have one thousand different deep understandings, there will be no controversy when comes to the border artistic themes, such as politics vs social order, religions vs realism, histories vs nature.

We claim that computer vision should be able to translate an art work to a message, saying, understand the theme of a painting. Our previous work have provided a large art work dataset for research and the learning of the common properties between photos and art works may also help. This will be a totally new application and we believe the community of computer vision will benefit a lot from this research.

6.3 Conclusions

I will now finish this thesis by listing all the conclusions I have reached, ranked based on importance, from high to low.

1. Models of visual objects should not be premised, even tacitly, on photo-real appearance or indeed on any particular depictive style at all. Rather, visual object models should be based on quasi-invariant properties of the objects in a class.

2. Structure is an important information to model the visual object class. Evidence in Chapter 4 shows an object class can be characterised by the qualitative shape of object parts and their structural arrangement. Results in Chapter 5 shows that a model with inter-nodes relations performs better than a star-like graph such as DPM.

3. Multi-label nodes are useful representations in coping with features that exhibit very wide, possibly discontinuous distributions and experiments in Chapter 5 show it is better than using a monolithic model to capture such wide variations.

4. Segmented regions can be categorised as some primitive shapes such as 'triangle', 'square', or 'circle' at a higher rate than randomly generated regions, upon given appropriate descriptors and well-design classifers : see Chapter 3.

Our final conclusion is then: computer vision will be definitely benefit both theoretically and in practically, if the question of recognition regardless of depiction is more fully understood. This thesis just a start of this new area and it will be glad to see more computer vision people researching based on our work.

APPENDIX A

CHOOSING THE ORDER AND RESOLUTION OF ZERNIKE MOMENTS

The results in the main section of this paper use Zernike moments of order 6, re-sampled onto a discrete 50×50 regular grid of pixels. This choice is justified by extensive experiments, which are described in this appendix.

We used 3 databases and 3 segmentation algorithms (region producers), so we can generate 9 region sets. In addition we are also able to create random regions. During these experiments we used two grid resolutions, 100^2 and 50^2. All regions taken from real images were subject to a whitening transform and re-sampling onto a grid of appropriate size, as explained in Sect 3.2.3. Random regions were generated directly at the correct resolution. We now have 18 sets of regions, harvested from images, re-sampled at 2 resolutions; and randomly generated regions, which is the same set but resized at 2 resolutions. We used 8 different orders of Zernike moment, specifically $\{4, 6, 8, 10, 12, 15, 30\}$. In each case we classified using the normalised form as described in section 3.2.4; that use the absolute value of each element (which removes orientation dependence) and scale by the first element (which allows for different number of binary points). This was followed by projection into an eigenframe using PCA; we retained 97% of the eigenenergy.

The next step is clustering. We initiate clustering using meanshift, with bandwidth automatically set as in section 3.2.5. We complete clustering using agglomerative clustering described in section 3.2.5.

Table A.1 shows the order of Zernike moment used, the full dimension of the Zernike feature space, the reduced PCA dimension, the bandwidth, and the percentage of regions we were able to classify. The table reports mean figures for the reduced dimension (rounded to the nearest integer) and for the faction of regions classified. The actual fraction of regions classified in each particular case can be read from Figure A-1. It can

Z-order	4	6	8	10	12	15	30
Dimension	14	27	44	65	90	135	495
Reduced-dimension	6	11	18	26	35	66	2
regions classified	0.60	0.63	0.64	0.66	0.67	0.67	1

Table A.1: *Zernike moment order (independent variable), dimension of raw feature space, reduced PCA dimension when keeping* 97% *of the eigenenergy, and fraction of regions classified; the latter two being averaged over all nine cases (see text).*

be seen that the worst case result occurs when the Bath images (of natural scenes) is segmented using MSER, where only about 1/3 of regions are classifiable – but compared to random this is a high value. The highest fraction observed is 1, which occurs for all cases whenever the Zernike moment order rises above about 30. This is an anomaly threat which requires an explanation.

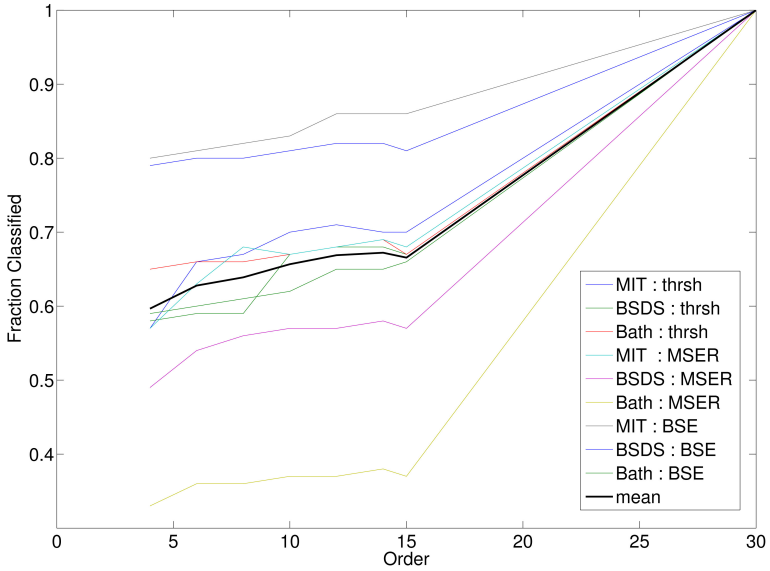


Figure A-1: *The fraction of significant shapes of different datasets with different segmentors on different Zernike moment order. The icon pixel size is* $50 * 50$.

These results show that the fraction of classifiable regions is stable for many orders of Zernike moments (less than 15). This stability is clearly visible in Figure A-1, which plots the fraction of regions classified as a function of Zernike moment order.

Unacceptable results are obtained if the moment is too high (30 or more). In these cases the PCA dimension falls dramatically, and all regions are clustered into a single class. An explanation of this result will be given in the following section. Interestingly, the answer is not the resolution of the re-sampling grid: we obtained nearly identical results for the 100^2 case.

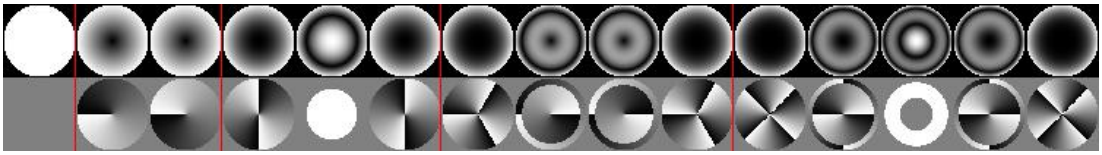


Figure A-2: *Zernike Polynomials, for order 0,1,2,3,4; arranged into groups. Top row shows the absolute value of the polynomial, lower row is the complex phase.*

A.1 Anomalous Results for High Order Zernike Moments

In the above experiments, we observed that when the order of moment is bigger than 30 or more, the data will be crowded into one or two dimension after using PCA. This requires an explanation.

Zernike moments up to order 4 are shown in Figure A-2. Each order, n, has $n+1$ basis, for $m = -n, -n+2, ...n-2, n$. Moments of even order have some component such that $m = 0$, which are easily identified in Figure A-2 because the complex phase appears as a flat white colour, showing it to be constant. We will explain the sudden fall in PCA dimension using only these components, because they are representative of the order as whole.

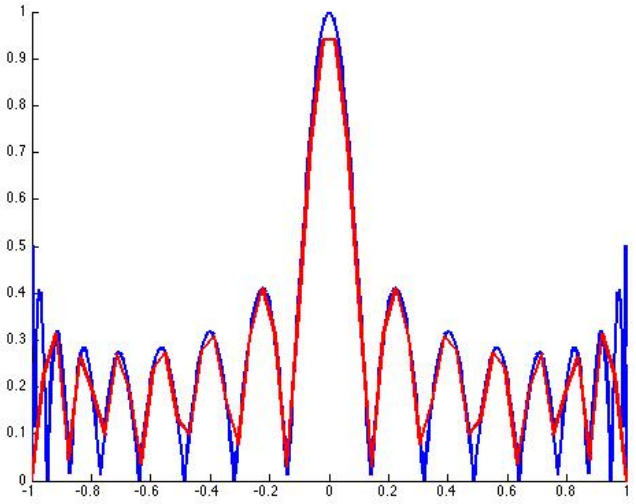


Figure A-3: *Cross-section through $n = 20, m = 0$ at resolution 500 (blue), and 50 (red). The effect of aliasing is clearly visible close to the edges of the polynomial (close to ± 1).*

As seen in Figure A-3, a cross-section through these basis functions shows that are rectified waveforms. Comparison with Figure A-2 confirms that the frequency of oscillation rises with distance from the origin at 0. This results in aliasing that is most noticeable at the perimeter of the disc – clearly visible in Figure A-3. On a discretely sampled plane, these basis functions alias badly close to the perimeter of the unit disc. This is clearly seen in Figure A-4, which reconstructs two simple shapes from

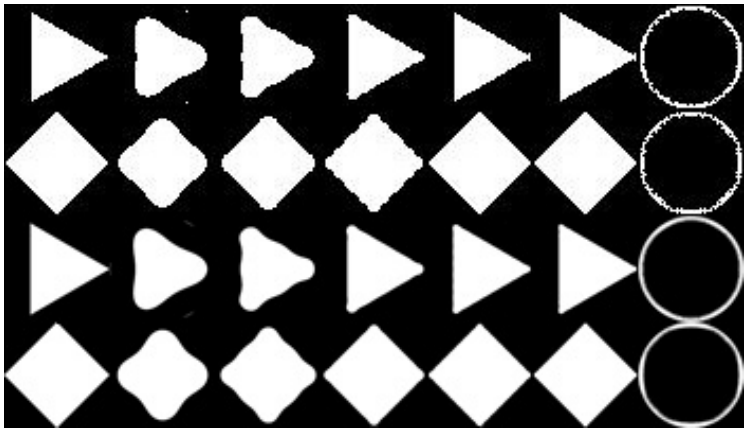


Figure A-4: *Reconstruction icons of the Zernike moment by using moments* $6, 10, 20, 30, 40, 50$. *The original regular shapes (triangle and square) are shown on the left. The top two rows are* 50^2 *image, the bottom two* 500^2 *images. When the moment order is bigger than 40, the aliasing occurs; then all shapes are reconstructed with a surrounding ring.*

Zernike moments of increasing order. At high orders the aliasing effects dominate the reconstruction – both are reconstructed as a ring like object. This explains why we observe a fall in the PCA dimension for high order Zernike moments: aliasing means that they encode ring-like objects.

This conclusion is re-enforced when we consider RMS reconstruction error.

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n} (\widehat{y_i} - y_i)^2}{n}} \tag{A.1}$$

in which $\widehat{y_i}$ is the reconstruction shape pixel, y_i is the original shape pixel and n is the number of pixels of this shape. Figure A-5 shows the reconstruction error for a triangle

Figure A-5: *The error as a fraction of original shape (a triangle) with the increasingly moment order. The red curve is the 100*100 resolution one and the green curve is 200*200.*

on two grid sizes (100^2 and 200^2). This is seen to fall as the order of the moment rises from 1 to a minimum at about 45, after which the error rises sharply.

The exact location of the minimum depends on the shape of the region – triangles are quite robust. Repeating the experiment on 10808 regions harvested from the MIT database using thresholding segmentation method, we found the minimum occurs as low as order 10 for some shapes, and as high as 59 for others.

A.2 Appendix Conclusion

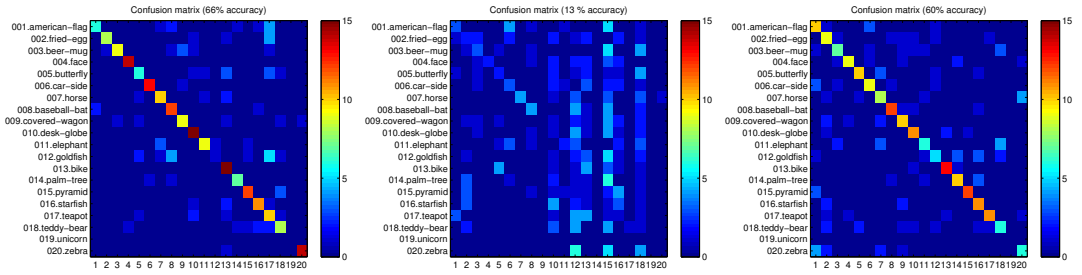
These results show that choosing a Zernike moment of 6 and grid resolution of 50^2 make little difference to our classifier. The choice represents a balance between computational efficiency representational accuracy.

APPENDIX B

CONFUSION MATRIX FOR EACH TEST CASE IN CHAPTER 4

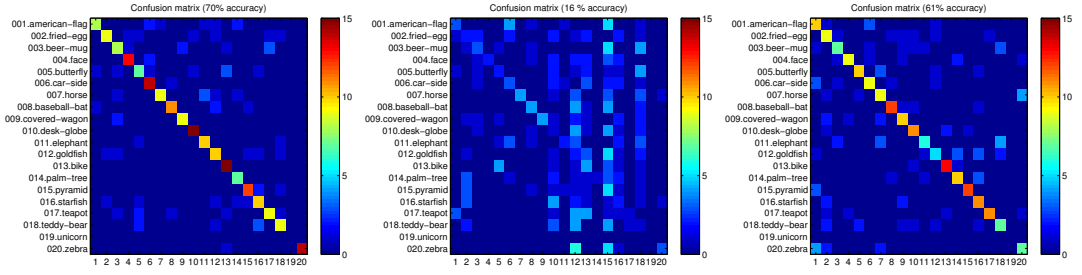
B.1 Training on Photos Alone

In this experiment, only photos (real object photos) are trained, with different numbers. And we test on photos and artwork separately. Three methods are used.



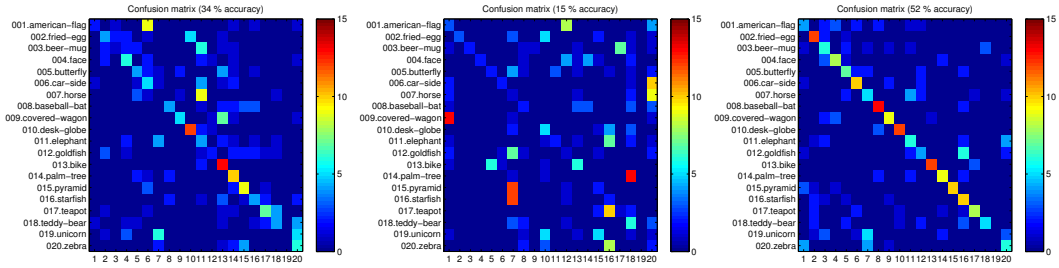
(a) (b) (c)

Figure B-1: *Train on 3 photos each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



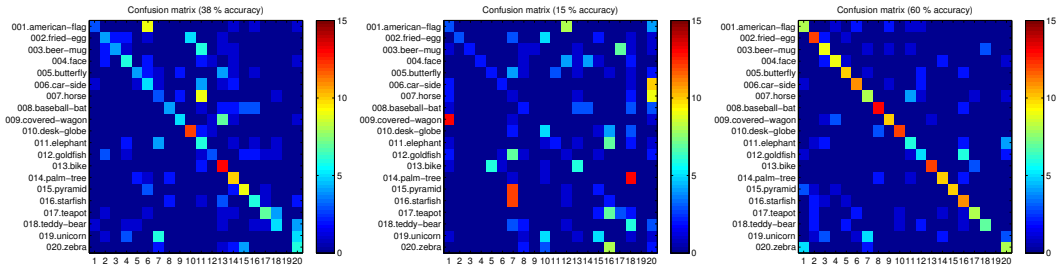
(a) (b) (c)

Figure B-2: *Train on 5 photos each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



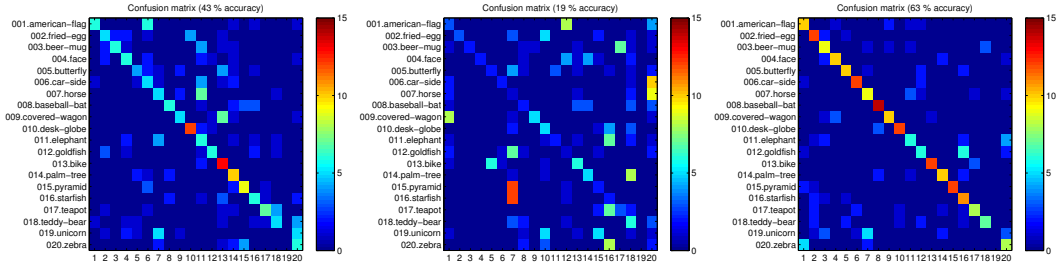
(a) (b) (c)

Figure B-3: *Train on 3 photos each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



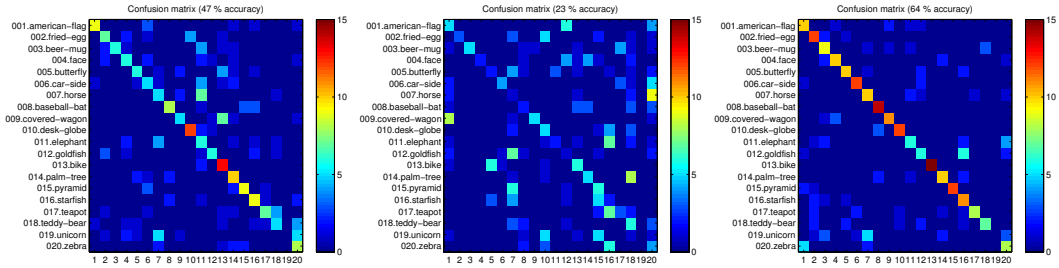
(a) (b) (c)

Figure B-4: *Train on 5 photos each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



(a) (b) (c)

Figure B-5: *Train on 8 photos each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*

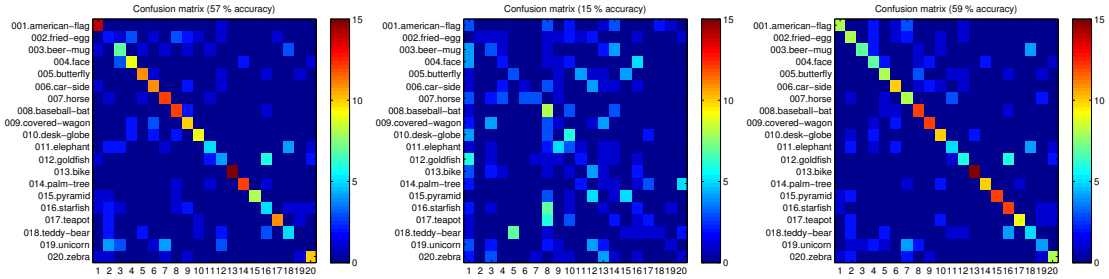


(a) (b) (c)

Figure B-6: *Train on 10 photos each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*

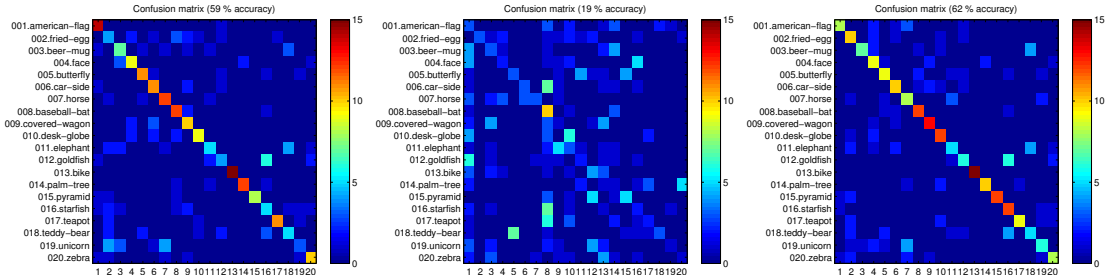
B.2 Training on Artwork Alone

In this experiment, only Artwork (paintings, line drawings .etc) are trained, with different numbers. And we test on photos and artwork separately. Three methods are used.



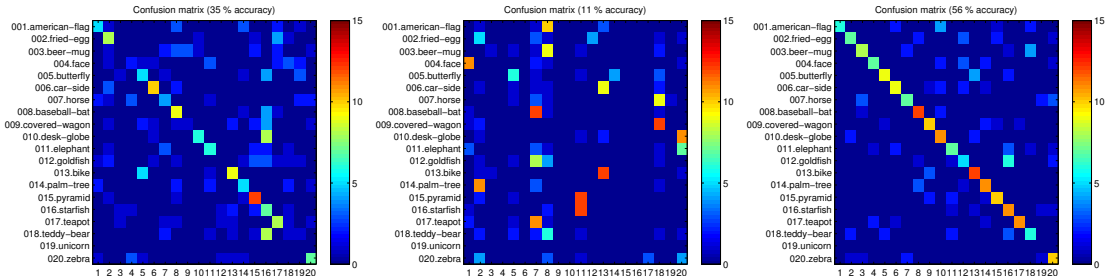
(a) (b) (c)

Figure B-7: *Train on 3 artwork each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



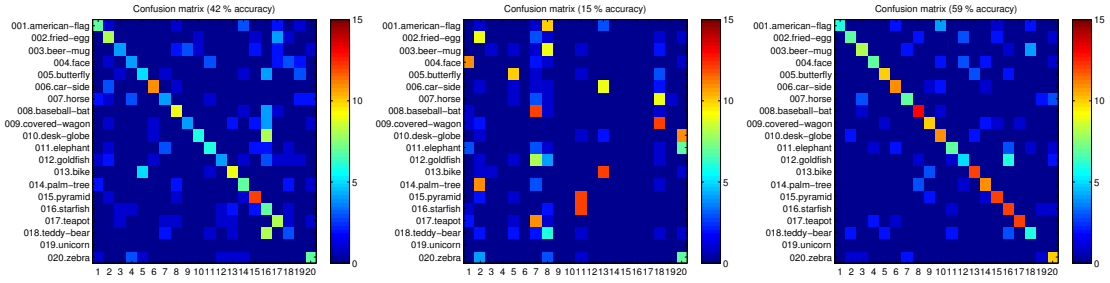
(a) (b) (c)

Figure B-8: *Train on 5 artwork each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



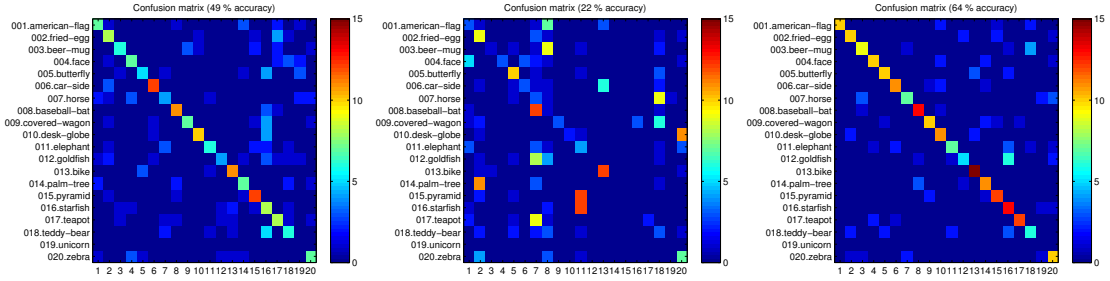
(a) (b) (c)

Figure B-9: *Train on 3 artwork each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



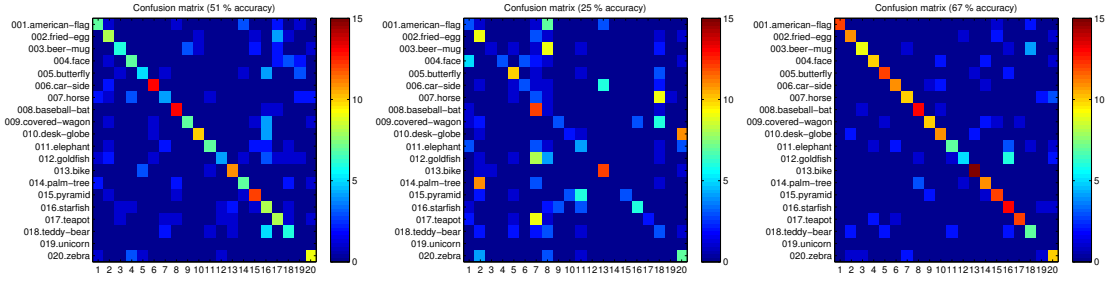
(a) (b) (c)

Figure B-10: *Train on 5 artwork each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



(a) (b) (c)

Figure B-11: *Train on 8 artwork each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*

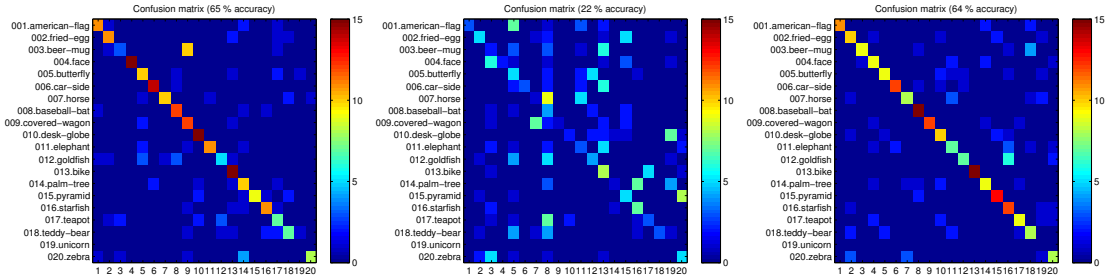


(a) (b) (c)

Figure B-12: *Train on 10 artwork each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*

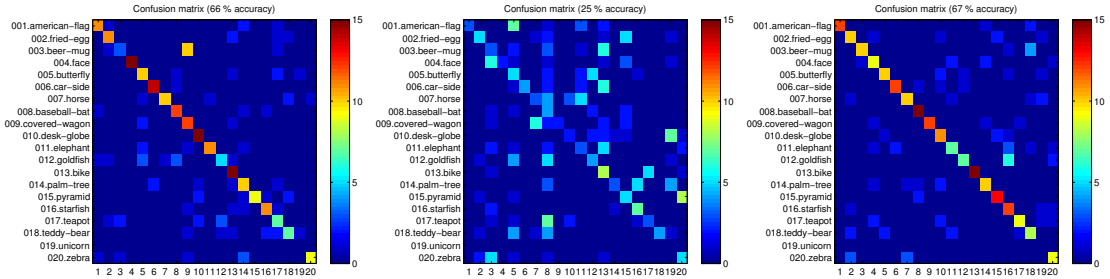
B.3 Training a Mixture

In this experiment, both artwork (paintings, line drawings .etc) and photos are trained as a mixture, with different numbers. And we test on photos and artwork separately. Three methods are used.



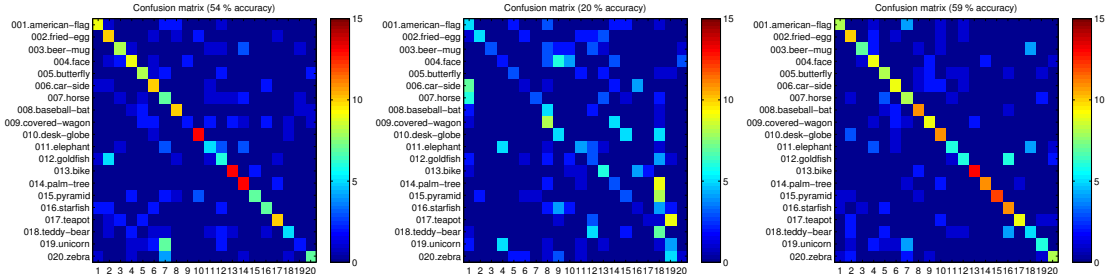
(a) (b) (c)

Figure B-13: *Train on 3 artwork+3 photos each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



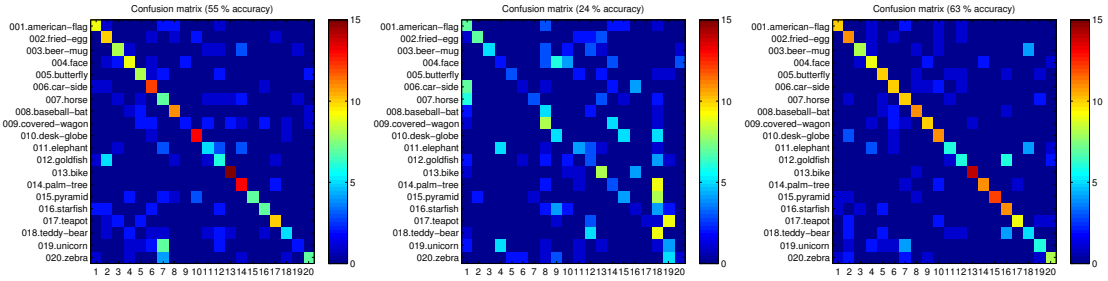
(a) (b) (c)

Figure B-14: *Train on 5 artwork+5 photos each class, test on 15 photos each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



(a) (b) (c)

Figure B-15: *Train on 3 artwork+3 photos each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*



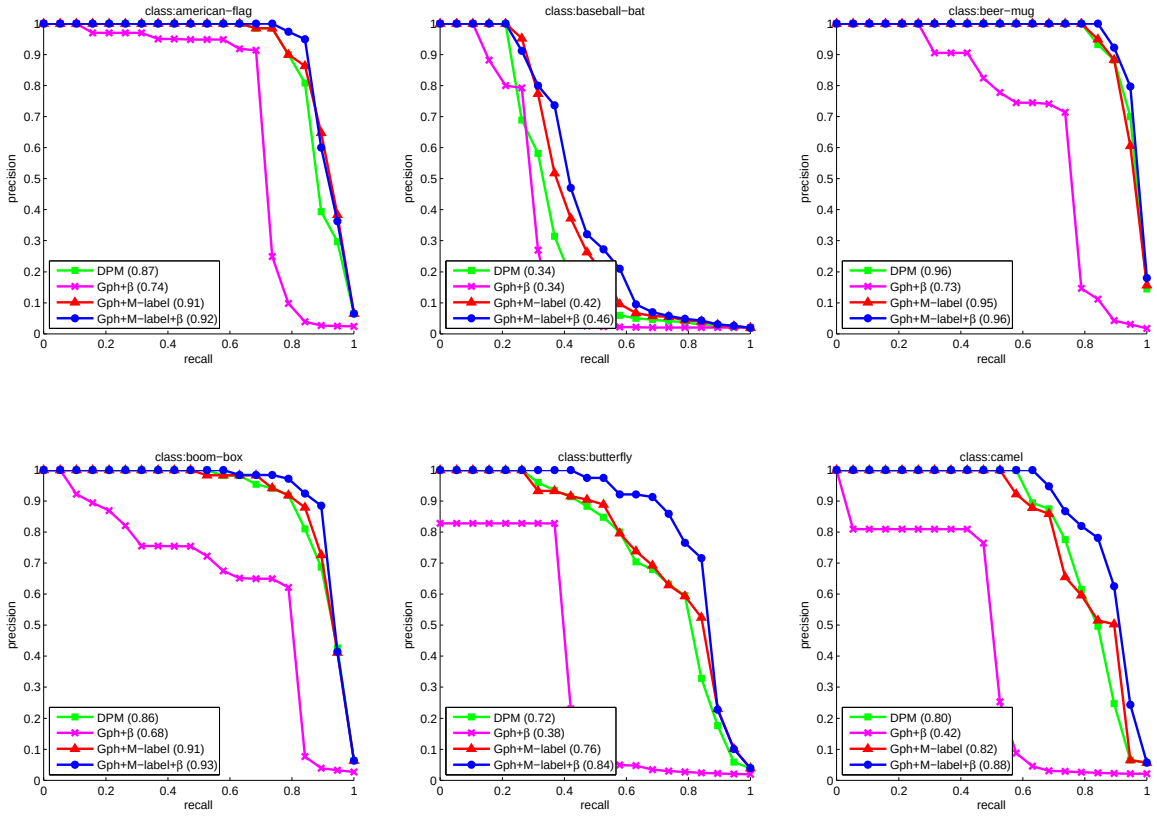
(a) (b) (c)

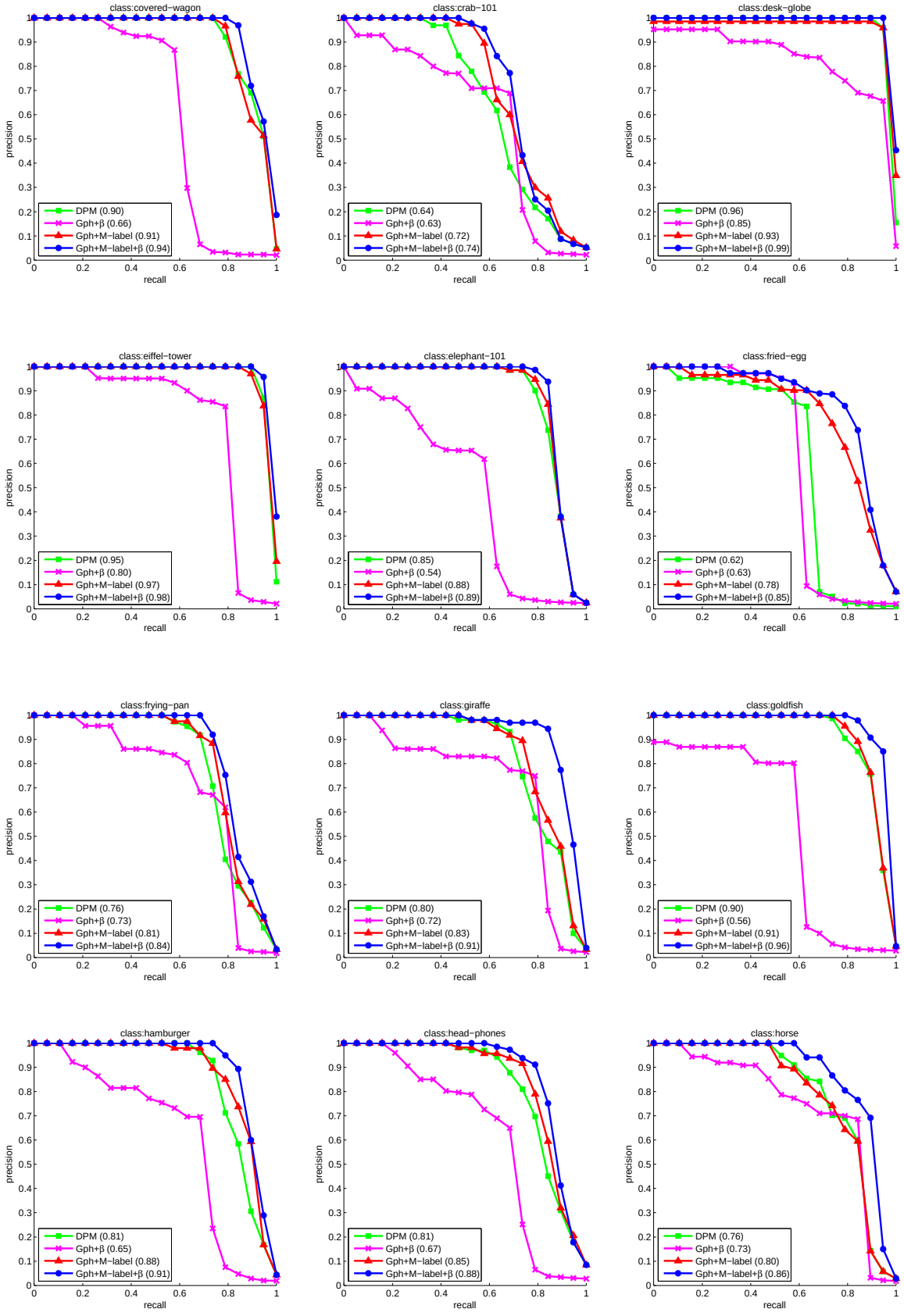
Figure B-16: *Train on 5 artwork+5 photos each class, test on 15 artwork each class, using* (a)*: Dense SIFT [147].* (b)*: Structure Only[167].* (c)*: Proposal Method*

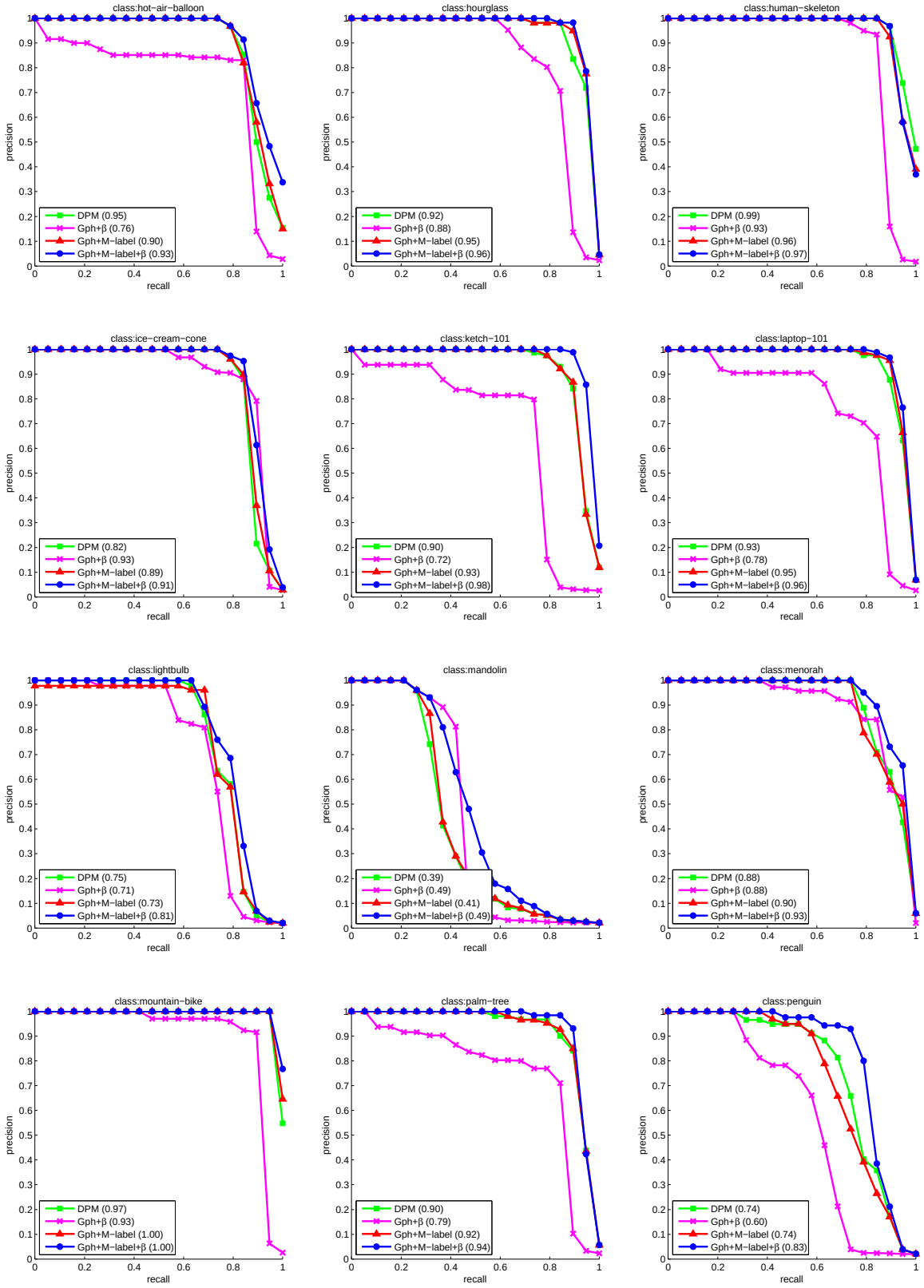
APPENDIX C

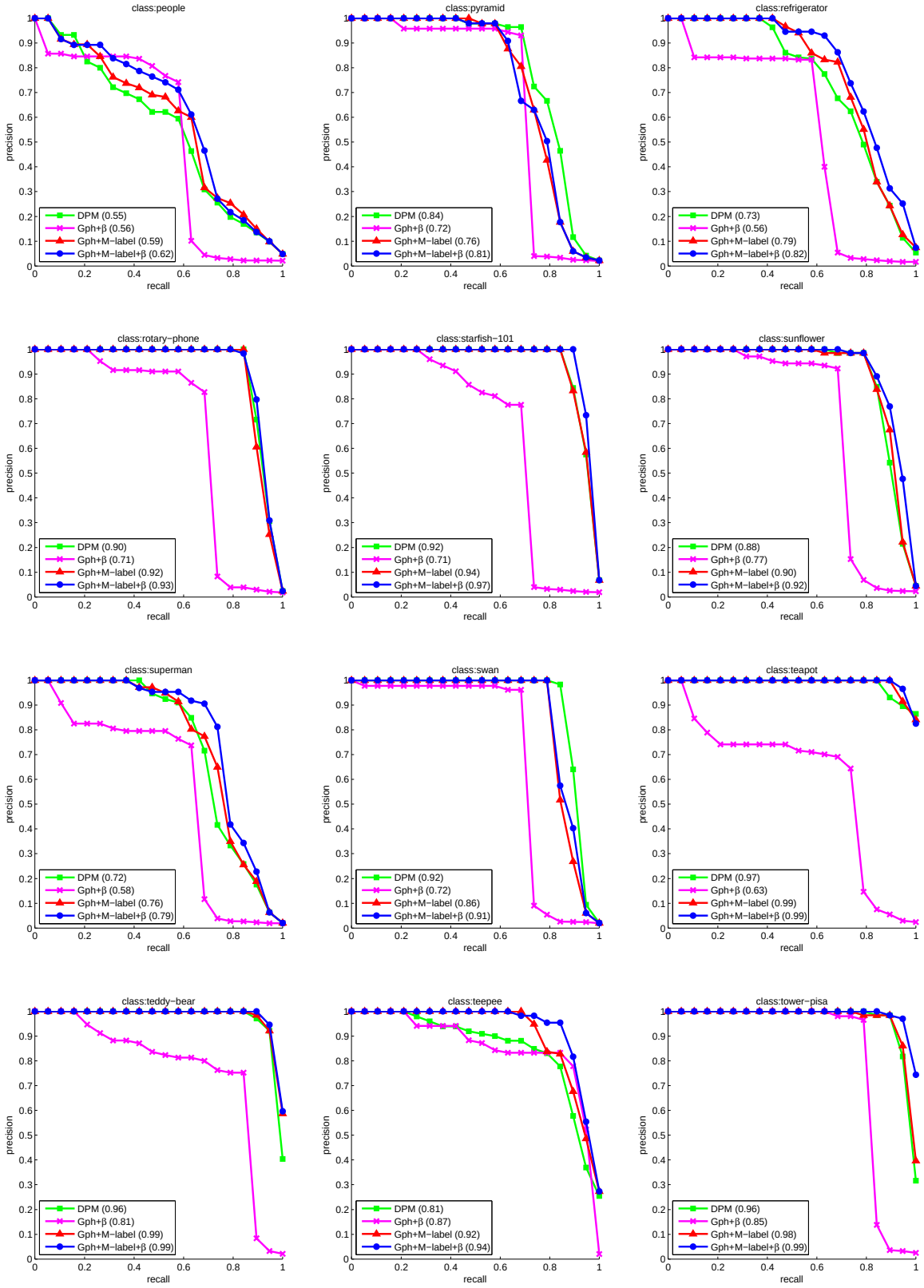
PRECISION AND RECALL CURVES FOR DETECTION ON PHOT-ART-50

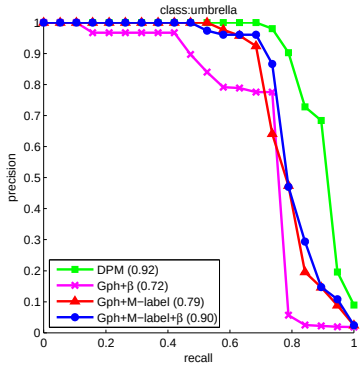
In this section, we show the Precision/Recall curves for models trained on the 50 categories of our cross-domain dataset. We show results for DPM, a single labeled graph model with learned β, our proposed multi-labeled model graph with and without learned β. In parenthesis we show the average precision score for each model.

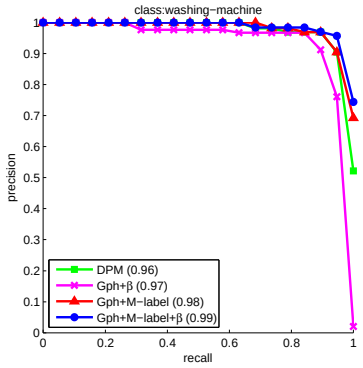


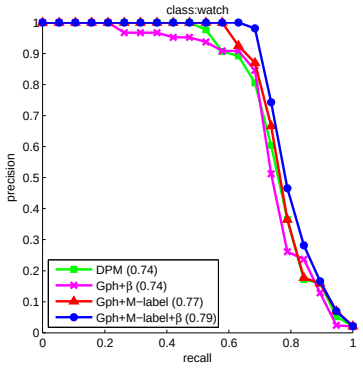

class:covered−wagon
DPM (0.90)
Gph+β (0.66)
Gph+M−label (0.91)
Gph+M−label+β (0.94)
class:crab−101
DPM (0.64)
Gph+β (0.63)
Gph+M−label (0.72)
Gph+M−label+β (0.74)
class:desk−globe
DPM (0.96)
Gph+β (0.85)
Gph+M−label (0.93)
Gph+M−label+β (0.99)
class:eiffel−tower
DPM (0.95)
Gph+β (0.80)
Gph+M−label (0.97)
Gph+M−label+β (0.98)
class:elephant−101
DPM (0.85)
Gph+β (0.54)
Gph+M−label (0.88)
Gph+M−label+β (0.89)
class:fried−egg
DPM (0.62)
Gph+β (0.63)
Gph+M−label (0.78)
Gph+M−label+β (0.85)
class:frying−pan
DPM (0.76)
Gph+β (0.73)
Gph+M−label (0.81)
Gph+M−label+β (0.84)
class:giraffe
DPM (0.80)
Gph+β (0.72)
Gph+M−label (0.83)
Gph+M−label+β (0.91)
class:goldfish
DPM (0.90)
Gph+β (0.56)
Gph+M−label (0.91)
Gph+M−label+β (0.96)
class:hamburger
DPM (0.81)
Gph+β (0.65)
Gph+M−label (0.88)
Gph+M−label+β (0.91)
class:head−phones
DPM (0.81)
Gph+β (0.67)
Gph+M−label (0.85)
Gph+M−label+β (0.88)
class:horse
DPM (0.76)
Gph+β (0.73)
Gph+M−label (0.80)
Gph+M−label+β (0.86)
precision
recall

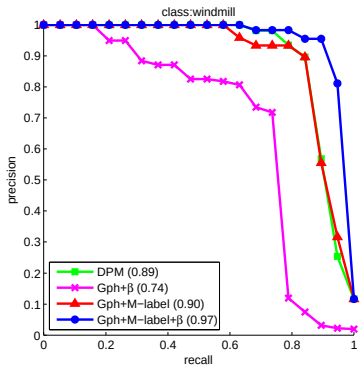

class:hot−air−balloon
DPM (0.95)
Gph+β (0.76)
Gph+M−label (0.90)
Gph+M−label+β (0.93)
class:hourglass
DPM (0.92)
Gph+β (0.88)
Gph+M−label (0.95)
Gph+M−label+β (0.96)
class:human−skeleton
DPM (0.99)
Gph+β (0.93)
Gph+M−label (0.96)
Gph+M−label+β (0.97)
class:ice−cream−cone
DPM (0.82)
Gph+β (0.93)
Gph+M−label (0.89)
Gph+M−label+β (0.91)
class:ketch−101
DPM (0.90)
Gph+β (0.72)
Gph+M−label (0.93)
Gph+M−label+β (0.98)
class:laptop−101
DPM (0.93)
Gph+β (0.78)
Gph+M−label (0.95)
Gph+M−label+β (0.96)
class:lightbulb
DPM (0.75)
Gph+β (0.71)
Gph+M−label (0.73)
Gph+M−label+β (0.81)
class:mandolin
DPM (0.39)
Gph+β (0.49)
Gph+M−label (0.41)
Gph+M−label+β (0.49)
class:menorah
DPM (0.88)
Gph+β (0.88)
Gph+M−label (0.90)
Gph+M−label+β (0.93)
class:mountain−bike
DPM (0.97)
Gph+β (0.93)
Gph+M−label (1.00)
Gph+M−label+β (1.00)
class:palm−tree
DPM (0.90)
Gph+β (0.79)
Gph+M−label (0.92)
Gph+M−label+β (0.94)
class:penguin
DPM (0.74)
Gph+β (0.60)
Gph+M−label (0.74)
Gph+M−label+β (0.83)
precision
recall

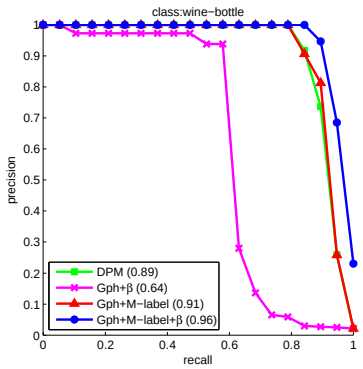

class:people
DPM (0.55)
Gph+β (0.56)
Gph+M−label (0.59)
Gph+M−label+β (0.62)
class:pyramid
DPM (0.84)
Gph+β (0.72)
Gph+M−label (0.76)
Gph+M−label+β (0.81)
class:refrigerator
DPM (0.73)
Gph+β (0.56)
Gph+M−label (0.79)
Gph+M−label+β (0.82)
class:rotary−phone
DPM (0.90)
Gph+β (0.71)
Gph+M−label (0.92)
Gph+M−label+β (0.93)
class:starfish−101
DPM (0.92)
Gph+β (0.71)
Gph+M−label (0.94)
Gph+M−label+β (0.97)
class:sunflower
DPM (0.88)
Gph+β (0.77)
Gph+M−label (0.90)
Gph+M−label+β (0.92)
class:superman
DPM (0.72)
Gph+β (0.58)
Gph+M−label (0.76)
Gph+M−label+β (0.79)
class:swan
DPM (0.92)
Gph+β (0.72)
Gph+M−label (0.86)
Gph+M−label+β (0.91)
class:teapot
DPM (0.97)
Gph+β (0.63)
Gph+M−label (0.99)
Gph+M−label+β (0.99)
class:teddy−bear
DPM (0.96)
Gph+β (0.81)
Gph+M−label (0.99)
Gph+M−label+β (0.99)
class:teepee
DPM (0.81)
Gph+β (0.87)
Gph+M−label (0.92)
Gph+M−label+β (0.94)
class:tower−pisa
DPM (0.96)
Gph+β (0.85)
Gph+M−label (0.98)
Gph+M−label+β (0.99)
precision
recall

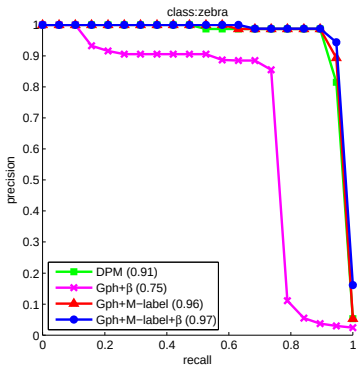

class:umbrella
precision
recall
DPM (0.92)
Gph+β (0.72)
Gph+M−label (0.79)
Gph+M−label+β (0.90)

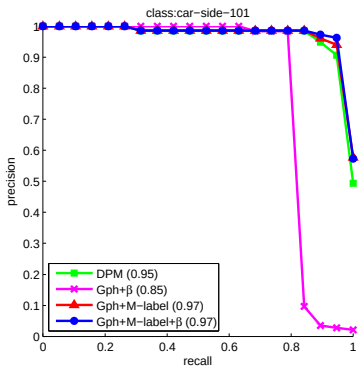

class:washing−machine
precision
recall
DPM (0.96)
Gph+β (0.97)
Gph+M−label (0.98)
Gph+M−label+β (0.99)

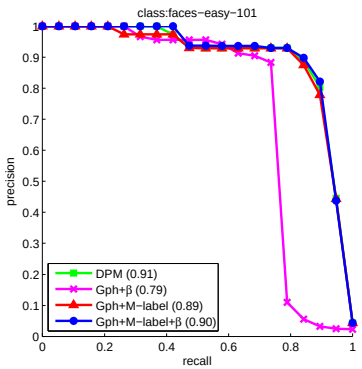

class:watch
precision
recall
DPM (0.74)
Gph+β (0.74)
Gph+M−label (0.77)
Gph+M−label+β (0.79)


class:windmill
precision
recall
DPM (0.89)
Gph+β (0.74)
Gph+M−label (0.90)
Gph+M−label+β (0.97)


class:wine−bottle
precision
recall
DPM (0.89)
Gph+β (0.64)
Gph+M−label (0.91)
Gph+M−label+β (0.96)


class:zebra
precision
recall
DPM (0.91)
Gph+β (0.75)
Gph+M−label (0.96)
Gph+M−label+β (0.97)

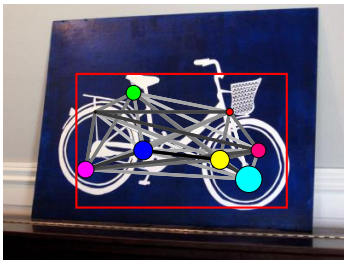
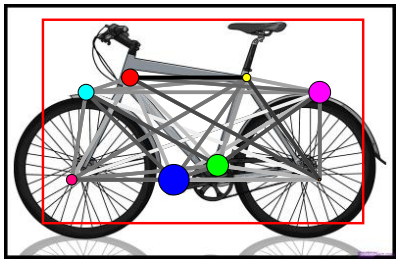
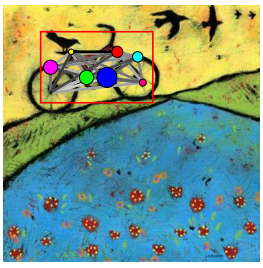
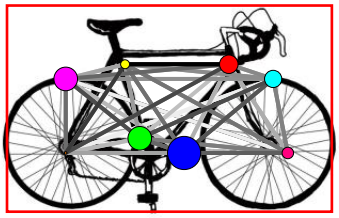
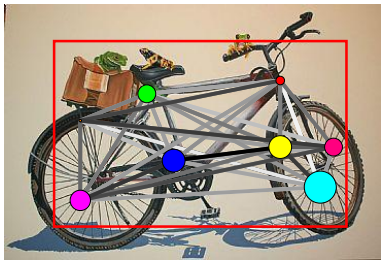
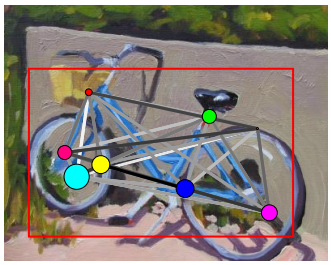
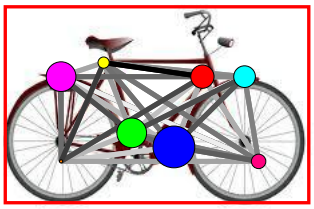
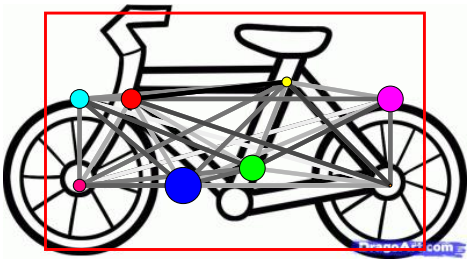
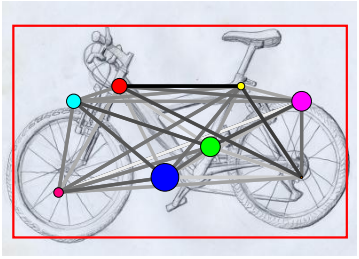

class:car−side−101
precision
recall
DPM (0.95)
Gph+β (0.85)
Gph+M−label (0.97)
Gph+M−label+β (0.97)

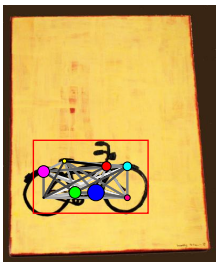
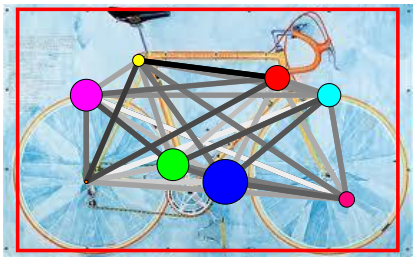
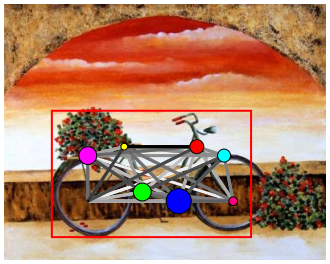
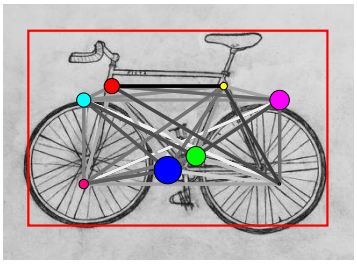
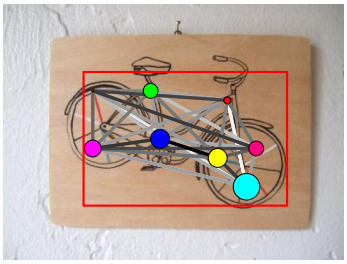
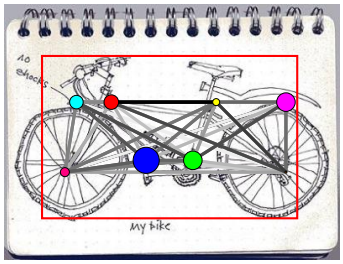
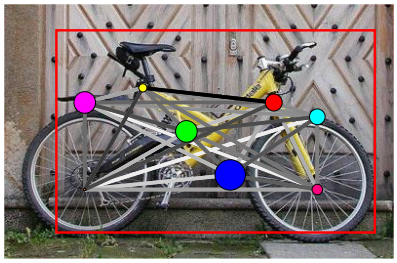

class:faces−easy−101
precision
recall
DPM (0.91)
Gph+β (0.79)
Gph+M−label (0.89)
Gph+M−label+β (0.90)

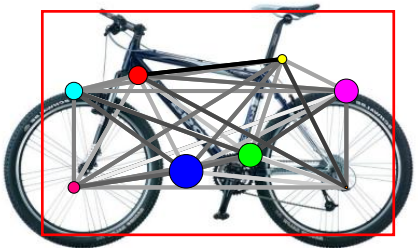
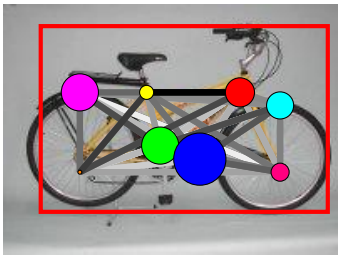
APPENDIX D

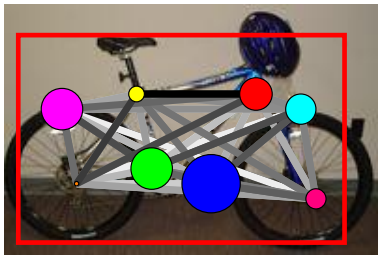
MORE DETECTION RESULTS

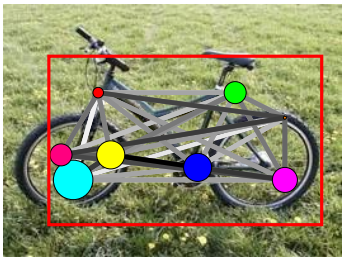
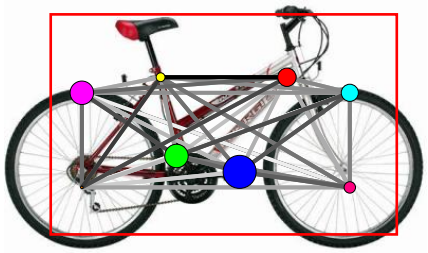
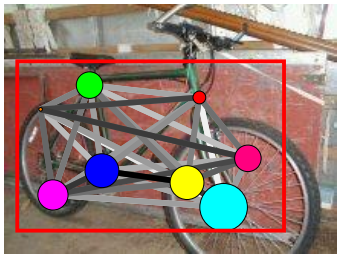
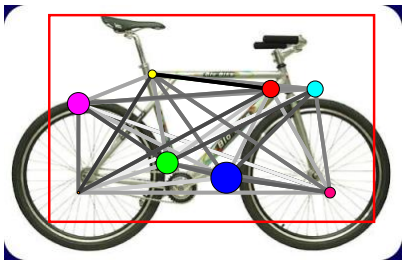
More detection results are shown in this appendix.

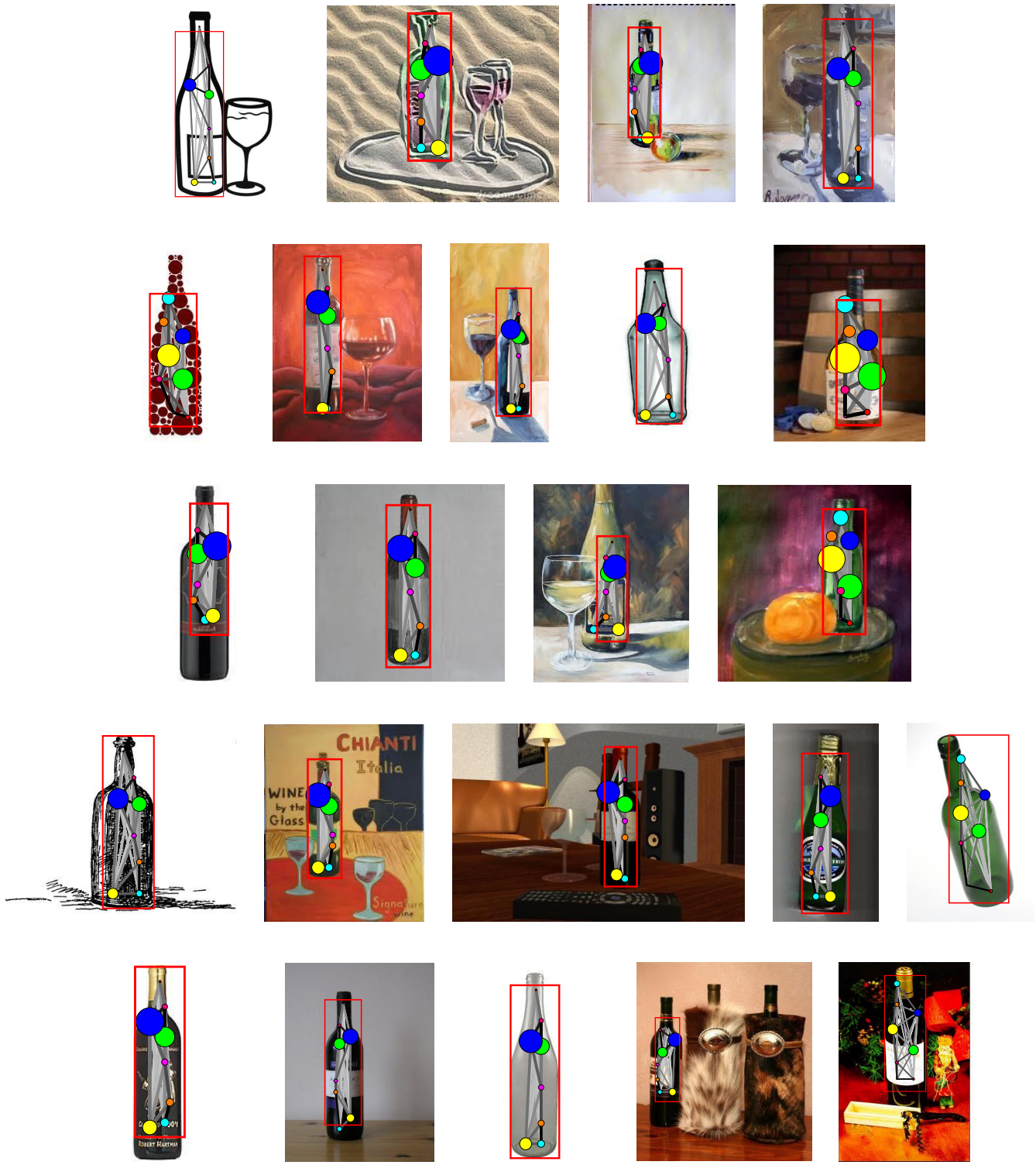










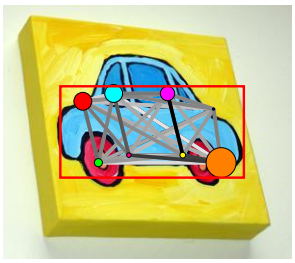
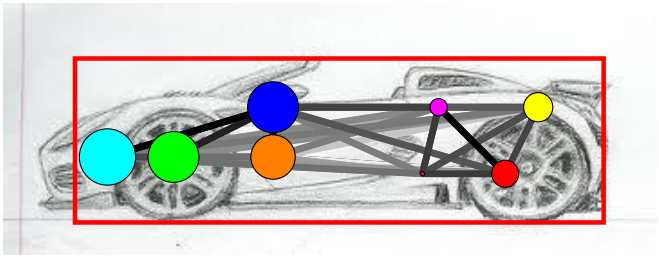






My bike


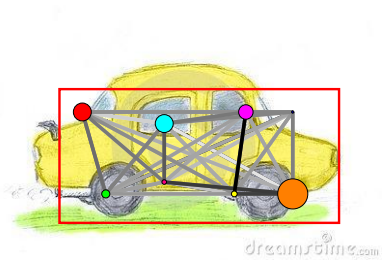
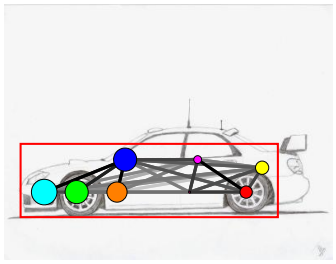
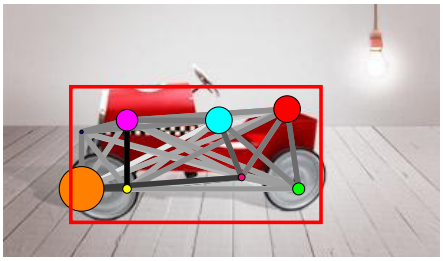
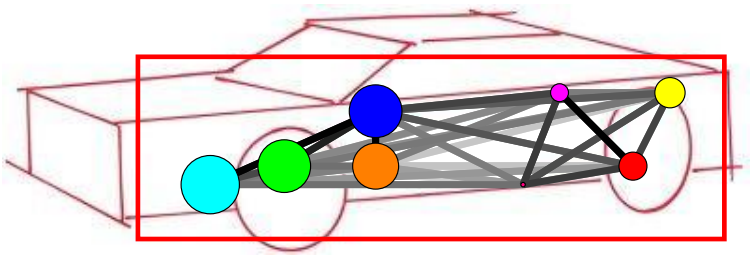
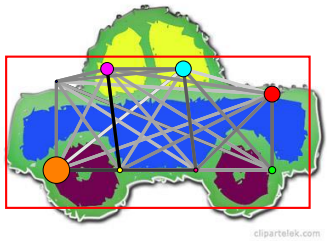
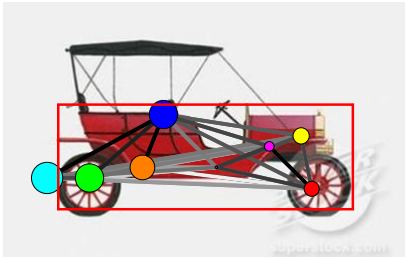
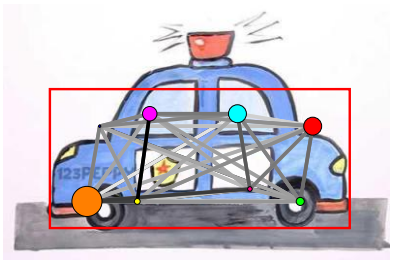
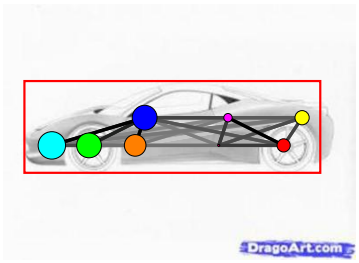
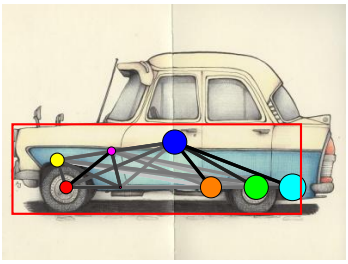
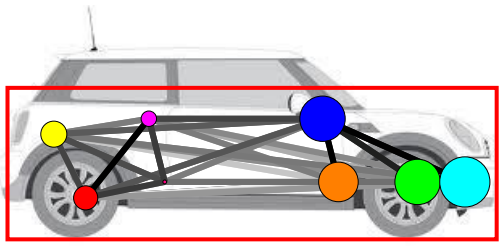
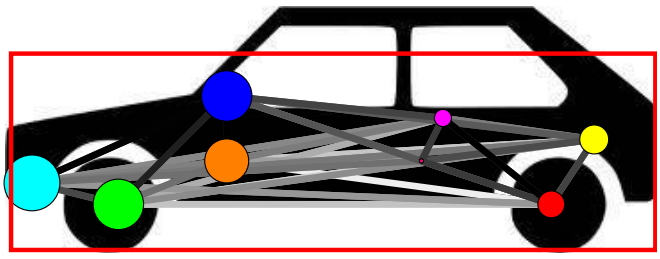
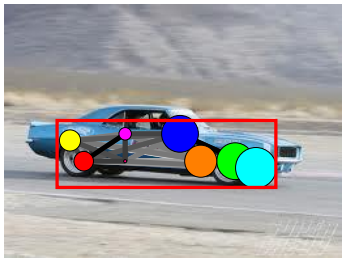
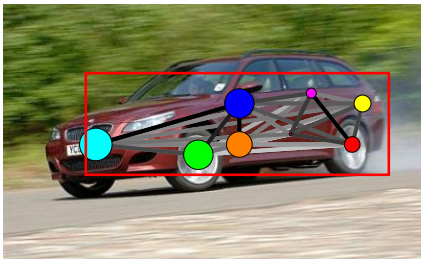
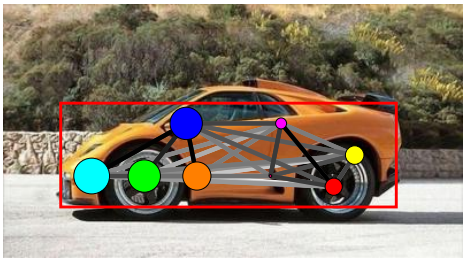



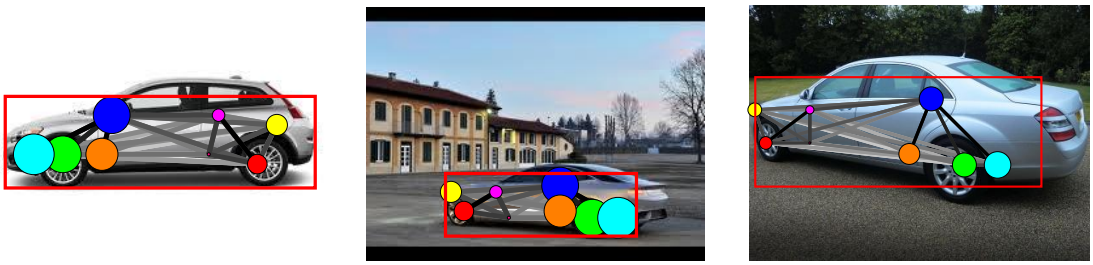


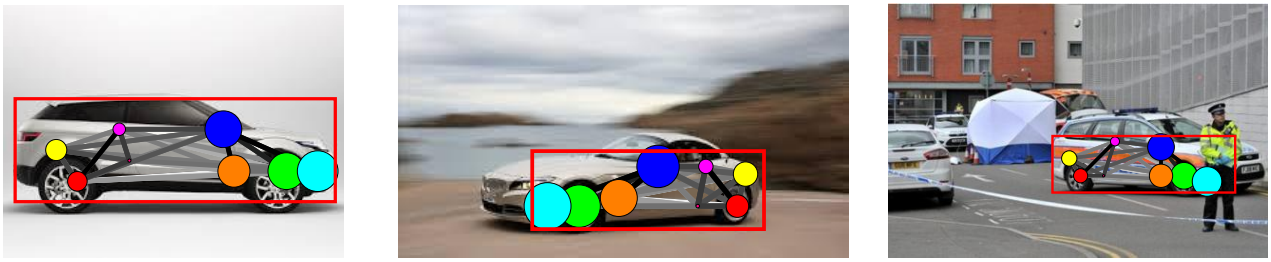





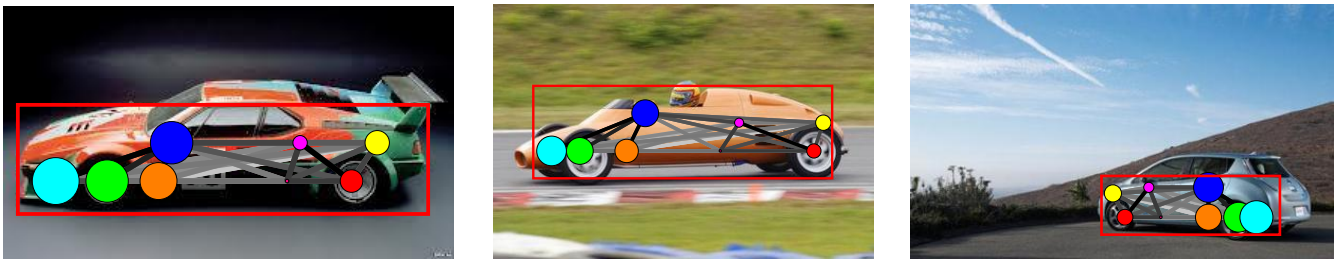


CHIANTI
Italia

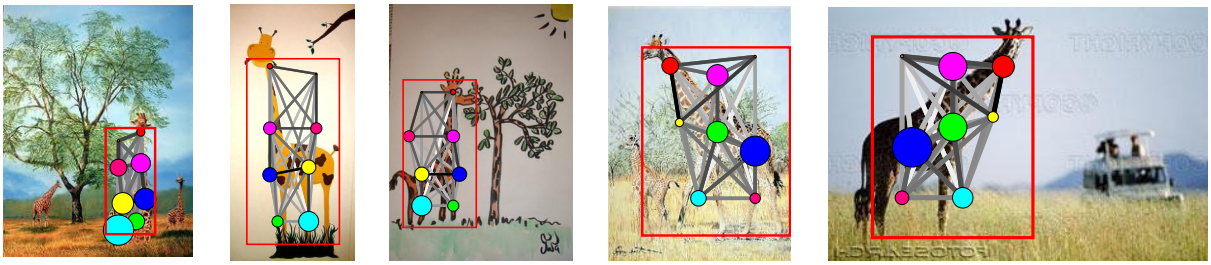



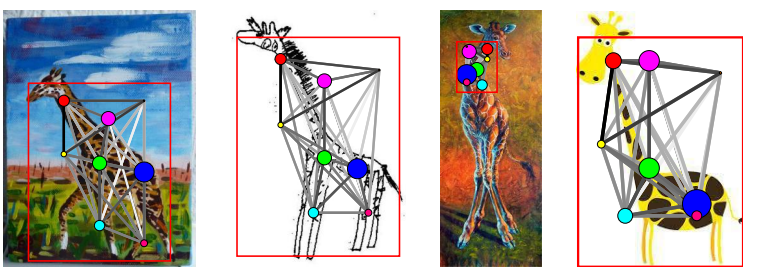
VROOM








DragoArt.com







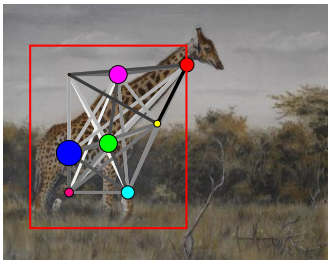
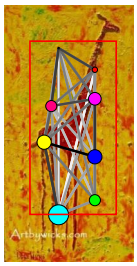
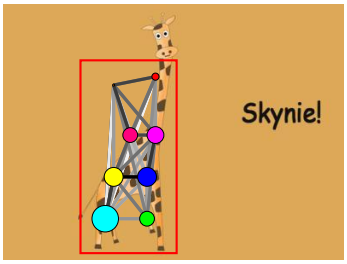


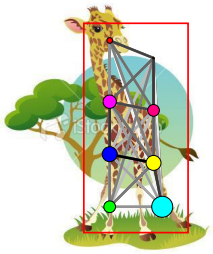
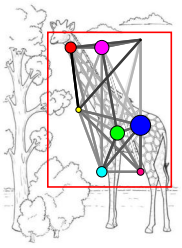
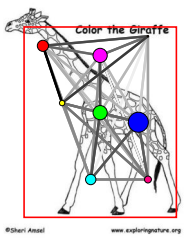


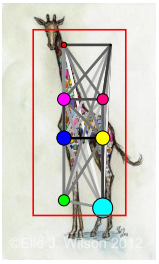
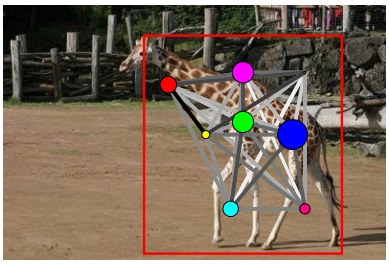
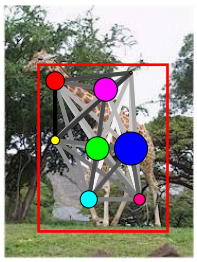
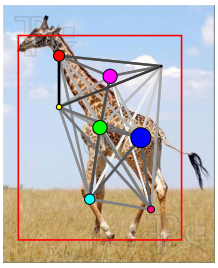






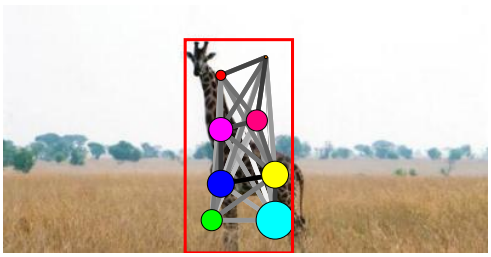
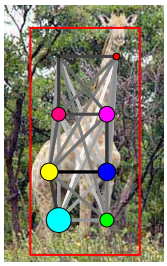
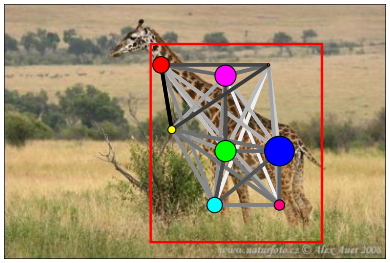
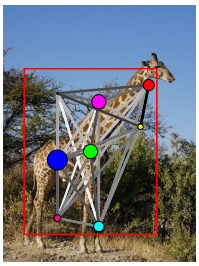
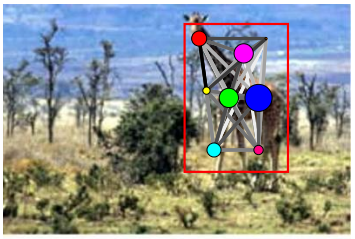
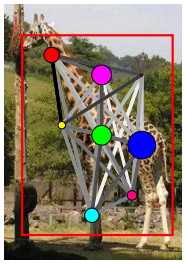
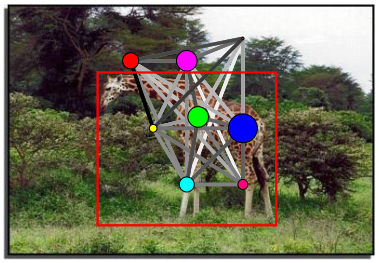
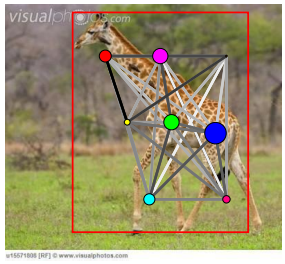
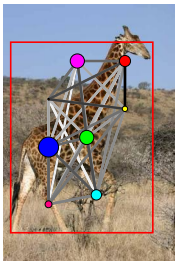
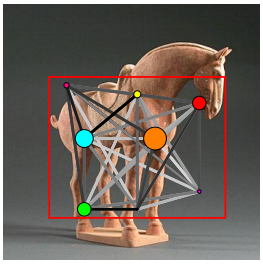
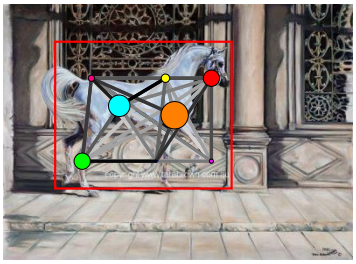
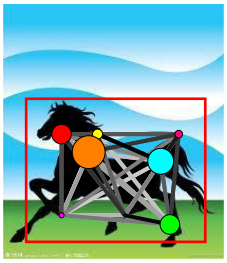
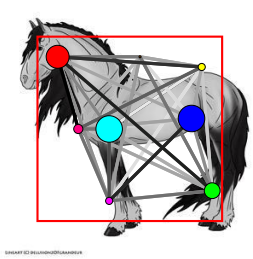
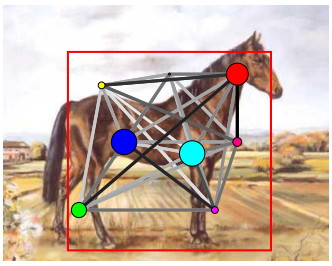
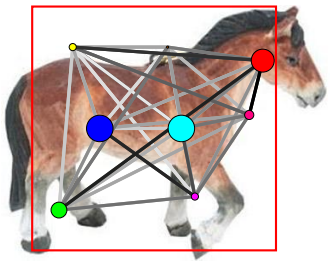
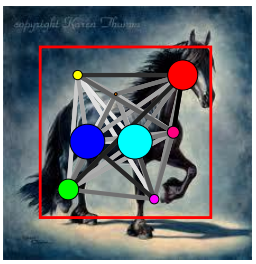
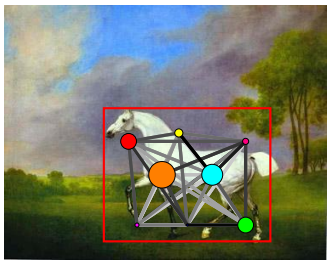
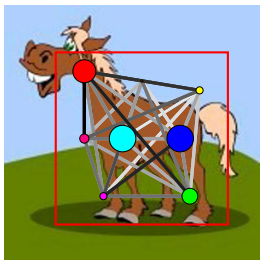
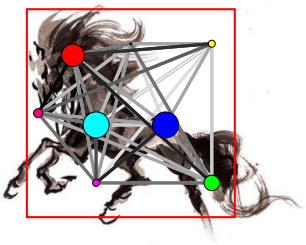
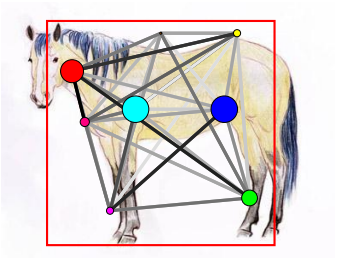
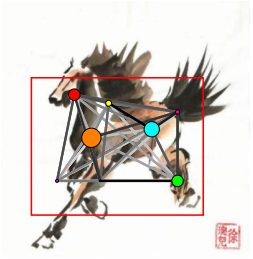
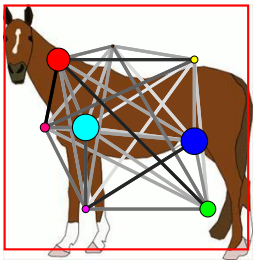
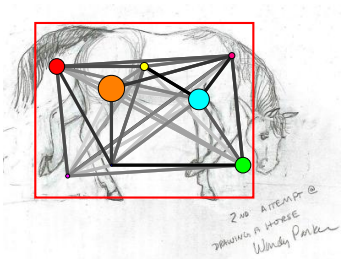
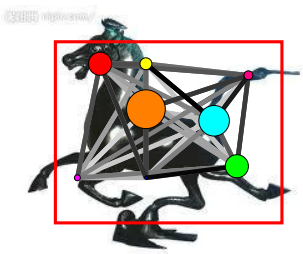
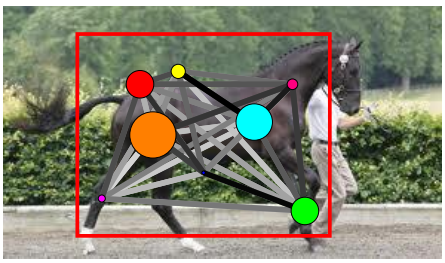
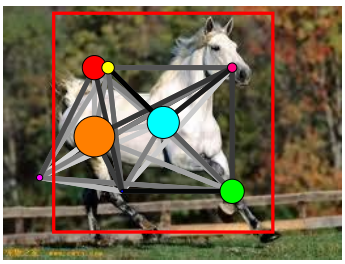
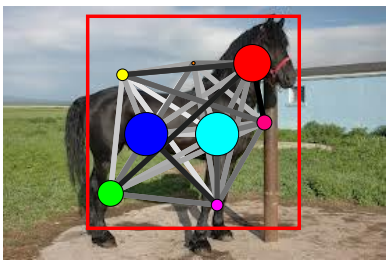
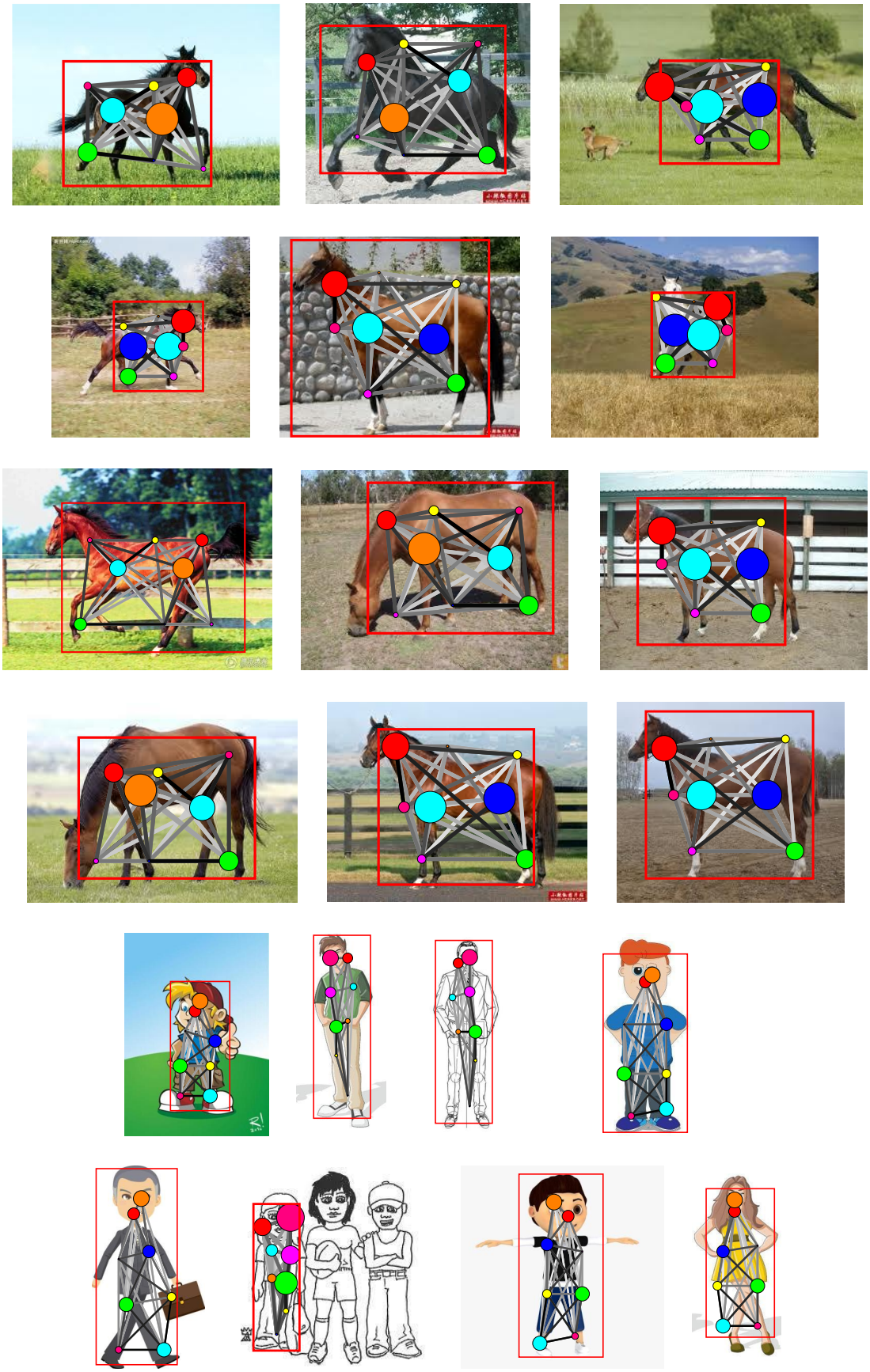
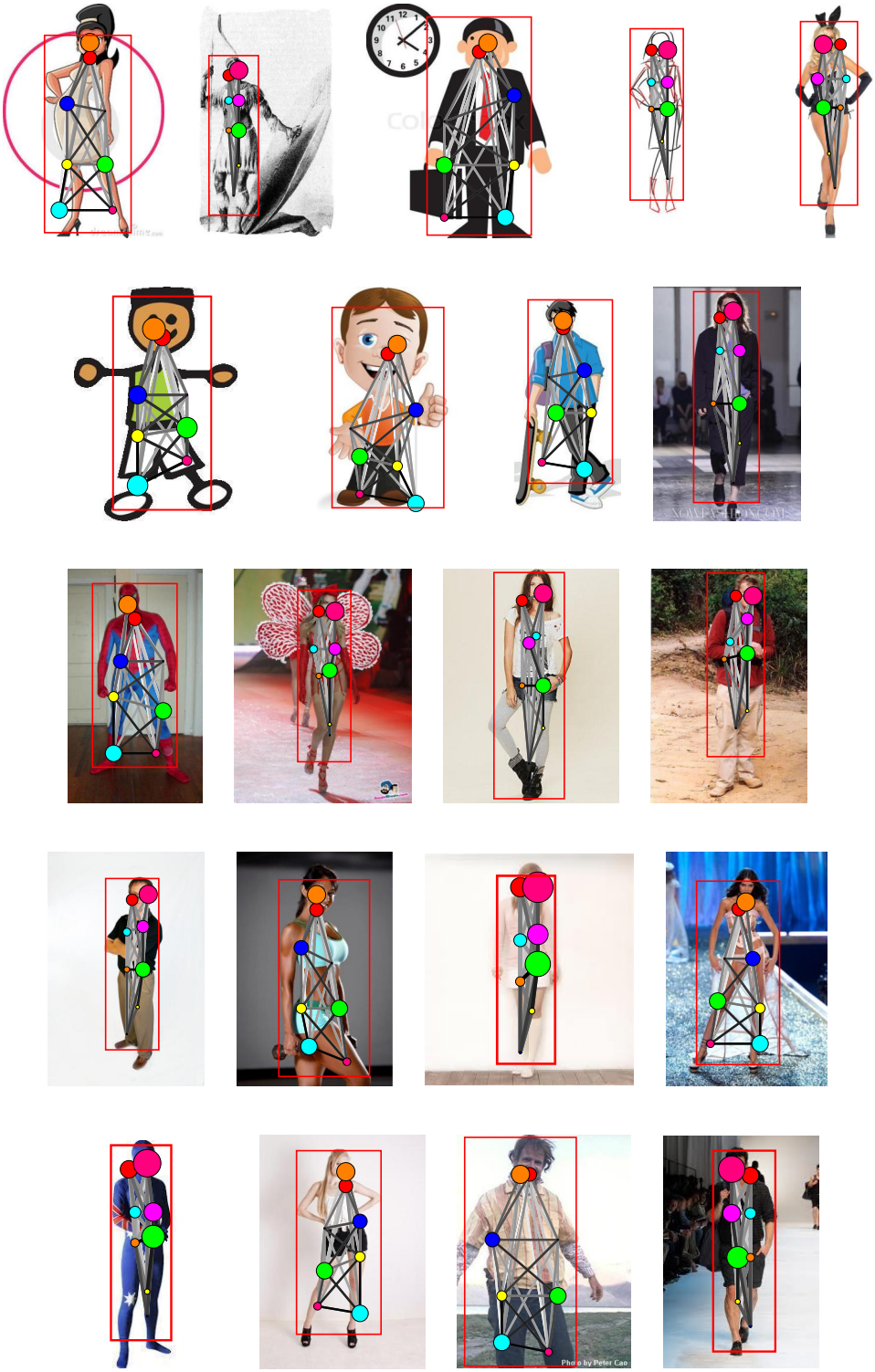






Photo by L.Romin/L.Dalton, Wildlife Reflections



















2ND ATTEMPT @
DRAWING A HORSE
Wendy Parker









BIBLIOGRAPHY

[1] N. Ahuja and S. Todorovic, *Learning the taxonomy and models of categories present in arbitrary images.*, in ICCV, 2007, pp. 1–8.

[2] N. AHUJA and S. Todorovic, *Connected segmentation tree-a joint representation of region layout and hierarchy*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[3] Y. Amit and A. Trouvé, *Pop: Patchwork of parts models for object recognition*, International Journal of Computer Vision, 75 (2007), pp. 267–282.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *From contours to regions: An empirical evaluation*, in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, June 2009, pp. 2294 –2301.

[5] M. Aubry, B. Russell, and J. Sivic, *Painting-to-3D model alignment via discriminative visual elements*, ACM Transactions on Graphics, (2013). Preprint, accepted for publication.

[6] A. Balikai, *Depiction invariant object matching, phd thesis*, 2012.

[7] A. Balikai and P. M. Hall, *Depiction inviariant object matching.*, in BMVC, 2012, pp. 1–11.

[8] A. Balikai, P. Rosin, Y.-Z. Song, and P. Hall, *Shapes fit for purpose*, in British Machine Vision Conference, 2008.

[9] E. Bart, E. Byvatov, and S. Ullman, *View-invariant recognition using corresponding object fragments*, in Computer Vision-ECCV 2004, Springer, 2004, pp. 152–165.

[10] S. Belongie, J. Malik, and J. Puzicha, *Shape matching and object recognition using shape contexts*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24 (2002), pp. 509–522.

[11] Y. Bengio, *Learning deep architectures for ai*, Foundations and trends® in Machine Learning, 2 (2009), pp. 1–127.

[12] A. C. Berg and J. Malik, *Geometric blur for template matching*, in IEEE International Conference on Computer Vision and Pattern Recognition, 2001.

[13] I. Biederman, *Recognition-by-components: a theory of human image understanding.*, Psychological review, 94 (1987), p. 115.

[14] A. Bosch, A. Zisserman, and X. Muñoz, *Scene classification via plsa*, in Computer Vision–ECCV 2006, Springer, 2006, pp. 517–530.

[15] A. Bosch, A. Zisserman, and X. Muoz, *Image classification using random forests and ferns*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007, pp. 1–8.

[16] H. Bunke, *Error-tolerant graph matching: A formal framework and algorithms*, in Advances in Pattern Recognition, A. Amin, D. Dori, P. Pudil, and H. Freeman, eds., vol. 1451 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1998, pp. 1–14.

[17] H. Bunke and S. Gunter, *Weighted mean of a pair of graphs*, Computing, 67 (2001), pp. 209–224.

[18] Y. Cao, C. Wang, L. Zhang, and L. Zhang, *Edgel index for large-scale sketch-based image search*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 761–768.

[19] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang, *Mindfinder: interactive sketch-based image search on millions of images*, in Proceedings of the international conference on Multimedia, ACM, 2010, pp. 1605–1608.

[20] G. Carneiro, N. P. da Silva, A. Del Bue, and J. P. Costeira, *Artistic image classification: an analysis on the printart database*, in Computer Vision–ECCV 2012, Springer, 2012, pp. 143–157.

[21] G. Cauwenberghs and T. Poggio, *Incremental and decremental support vector machine learning*, Advances in neural information processing systems, (2001), pp. 409–415.

[22] A. Chalechale, G. Naghdy, and A. Mertins, *Sketch-based image matching using angular partitioning*, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 35 (2005), pp. 28–41.

[23] Y. Chans, Z. Lei, D. P. Lopresti, and S.-Y. Kung, *Feature-based approach for image retrieval by sketch*, in Voice, Video, and Data Communications, International Society for Optics and Photonics, 1997, pp. 220–231.

[24] K. Chatfield, J. Philbin, and A. Zisserman, *Efficient retrieval of deformable shape classes using local self-similarities*, in Workshop on Non-rigid Shape Analysis and Deformable Image Alignment, ICCV, 2009.

[25] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, *Sketch2photo: internet image montage*, in ACM Transactions on Graphics (TOG), vol. 28, ACM, 2009, p. 124.

[26] M. Cho, K. Alahari, and J. Ponce, *Learning graphs to match*, in ICCV, 2013.

[27] T. Cootes, C. Taylor, D. Cooper, and J. Graham, *Active shape models-their training and application*, Computer Vision and Image Understanding, 61 (1995), pp. 38 – 59.

[28] T. F. Cootes, G. J. Edwards, C. J. Taylor, et al., *Active appearance models*, TPAMI, (2001).

[29] J. Coughlan, A. Yuille, C. English, and D. Snow, *Efficient deformable template detection and localization without user initialization*, CVIU, (2000).

[30] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, *Spatial priors for part-based recognition using statistical models*, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, IEEE, 2005, pp. 10–17.

[31] E. Crowley and A. Zisserman, *The state of the art: Object retrieval in paintings using discriminative regions*, in British Machine Vision Conference, 2014.

[32] E. J. Crowley and A. Zisserman, *In search of art*, in Workshop on Computer Vision for Art Analysis, ECCV, 2014.

[33] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, *Visual categorization with bags of keypoints*, in Workshop on statistical learning in computer vision, ECCV, vol. 1, 2004, pp. 1–2.

[34] N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection*, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, IEEE, 2005, pp. 886–893.

[35] M. R. Daliri and V. Torre, *Robust symbolic representation for shape recognition and retrieval*, Pattern Recognition, 41 (2008), pp. 1782 – 1798.

[36] A. Del Bimbo and P. Pala, *Visual image retrieval by elastic matching of user sketches*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19 (1997), pp. 121–132.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.

[38] T. Deselaers and V. Ferrari, *Global and efficient self-similarity for object classification and detection*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1633–1640.

[39] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, and S. Yan, *Subcategory-aware object classification*, in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 827–834.

[40] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, *Sketch-based image retrieval: Benchmark and bag-of-features descriptors*, IEEE Transactions on Visualization and Computer Graphics, 17 (2011), pp. 1624–1636.

[41] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, *Sketch-based shape retrieval*, ACM Trans. Graph. (Proc. SIGGRAPH), 31 (2012), pp. 31:1–31:10.

[42] G. Elidan, G. Heitz, and D. Koller, *Learning object shape: From drawings to images*, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, 2006, pp. 2064–2071.

[43] S. M. A. Eslami, N. Heess, C. K. I. Williams, and J. Winn, *The shape boltzmann machine: a strong model of object shape*, in International Journal of Computer Vision, 2013.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results*. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[45] L. Fei-Fei, R. Fergus, and P. Perona, *Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories*, Computer Vision and Image Understanding, 106 (2007), pp. 59–70.

[46] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, *Object detection with discriminatively trained part-based models*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32 (2010), pp. 1627–1645.

[47] P. F. Felzenszwalb and D. P. Huttenlocher, *Pictorial structures for object recognition*, International Journal of Computer Vision, 61 (2005), pp. 55–79.

[48] R. Fergus, P. Perona, and A. Zisserman, *Object class recognition by unsupervised scale-invariant learning*, in Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, IEEE, 2003, pp. II–264.

[49] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, *Unsupervised visual domain adaptation using subspace alignment*, in ICCV, 2013.

[50] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, *Groups of adjacent contour segments for object detection*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30 (2008), pp. 36–51.

[51] V. Ferrari, F. Jurie, and C. Schmid, *From images to shape models for object detection*, IJCV, (2010).

[52] M. Ferrer, E. Valveny, F. Serratosa, K. Riesen, and H. Bunke, *An approximate algorithm for median graph computation using graph embedding*, in Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE, 2008, pp. 1–4.

[53] S. Fidler, G. Berginc, and A. Leonardis, *Hierarchical statistical learning of generic parts of object structure*, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, IEEE, 2006, pp. 182–189.

[54] S. Fidler, M. Boben, and A. Leonardis, *Learning hierarchical compositional representations of object structure*, S. Dickinson, A. Leonardis, B. Schiele, and TM, editors, Object Categorization: Computer and Human Vision Perspectives, (2009), pp. 196–215.

[55] M. A. Fischler and R. Elschlager, *The representation and matching of pictorial structures*, Computers, IEEE Transactions on, C-22 (1973), pp. 67–92.

[56] P.-E. Forssen and D. Lowe, *Shape descriptors for maximally stable extremal regions*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, oct. 2007, pp. 1 –8.

[57] K. Fukushima, *Neocognitron: A hierarchical neural network capable of visual pattern recognition*, Neural networks, 1 (1988), pp. 119–130.

[58] D. Gavrila, *Multi-feature hierarchical template matching using distance transforms*, in Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on, vol. 1, 1998, pp. 439–444 vol.1.

[59] S. Ginosar, D. Haas, T. Brown, and J. Malik, *Detecting people in cubist art*, arXiv preprint arXiv:1409.6235, (2014).

[60] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, in Computer Vision and Pattern Recognition, 2014.

[61] B. Gong, K. Grauman, and F. Sha, *Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation*, in ICML, 2013, pp. 222–230.

[62] B. Gong, Y. Shi, F. Sha, and K. Grauman, *Geodesic flow kernel for unsupervised domain adaptation*, in CVPR, 2012, pp. 2066–2073.

[63] R. Gopalan, R. Li, and R. Chellappa, *Domain adaptation for object recognition: An unsupervised approach*, in IEEE International Conference on Computer Vision, vol. 0, 2011, pp. 999–1006.

[64] G. Griffin, A. Holub, and P. Perona, *Caltech-256 object category dataset*, (2007).

[65] C. Gu, P. Arbelaez, Y. Lin, K. Yu, and J. Malik, *Multi-component models for object detection*, in ECCV, 2012.

[66] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, *Recognition using regions*, in CVRP, 2009.

[67] F. Han and S.-C. Zhu, *Bottom-up/top-down image parsing with attribute grammar*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31 (2009), pp. 59–73.

[68] C. Harris and M. Stephens, *A combined corner and edge detector.*, in Alvey vision conference, vol. 15, Manchester, UK, 1988, p. 50.

[69] G. Hinton, S. Osindero, and Y.-W. Teh, *A fast learning algorithm for deep belief nets*, Neural computation, 18 (2006), pp. 1527–1554.

[70] M.-K. Hu, *Visual pattern recognition by moment invariants*, Information Theory, IRE Transactions on, 8 (1962), pp. 179 –187.

[71] R. Hu, M. Barnard, and J. P. Collomosse, *Gradient field descriptor for sketch based retrieval and localization*, in ICIP, 2010, pp. 1025–1028.

[72] R. Hu and J. Collomosse, *A performance evaluation of gradient field hog descriptor for sketch based image retrieval*, Computer Vision and Image Understanding, 117 (2013), pp. 790–806.

[73] Y. Huang, K. Huang, Y. Yu, and T. Tan, *Salient coding for image classification*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1753–1760.

[74] T. T.-n. Huang Kai-Qi, Ren Wei-Qiang, *A review on image object classification and detection*, Chinese Journal of Computers, 36 (2013).

[75] D. H. Hubel and T. N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, The Journal of physiology, 160 (1962), p. 106.

[76] W. Jia and S. McKenna, *Classifying textile designs using bags of shapes*, in ICPR, 2010.

[77] Y. Jia, *Caffe: An open source convolutional architecture for fast feature embedding*, 2013.

[78] X. Jiang, A. Munger, and H. Bunke, *An median graphs: properties, algorithms, and applications*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23 (2001), pp. 1144–1151.

[79] T. Joachims, T. Finley, and C.-N. J. Yu, *Cutting-plane training of structural svms*, Machine Learning, (2009).

[80] T. Judd, K. Ehinger, F. Durand, and A. Torralba, *Learning to predict where humans look*, in Computer Vision, 2009 IEEE 12th International Conference on, 29 2009-oct. 2 2009, pp. 2106 –2113.

[81] F. Jurie and B. Triggs, *Creating efficient codebooks for visual recognition*, in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 1, IEEE, 2005, pp. 604–610.

[82] W.-Y. Kim and Y.-S. Kim, *A region-based shape descriptor using zernike moments*, Signal Processing: Image Communication, 16 (2000), pp. 95 – 102.

[83] K. Koffka, *Principles of Gestalt psychology*, Routledge, 1935.

[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Advances in neural information processing systems, 2012, pp. 1097–1105.

[85] M. P. Kumar, P. Torr, and A. Zisserman, *Extending pictorial structures for object recognition*, in Proc. BMVC, 2004, pp. 81.1–81.10. doi:10.5244/C.18.81.

[86] S. Lazebnik, C. Schmid, and J. Ponce, *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, IEEE, 2006, pp. 2169–2178.

[87] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[88] B. Leibe, A. Leonardis, and B. Schiele, *Robust object detection with interleaved categorization and segmentation*, IJCV, (2008).

[89] M. Leordeanu, M. Hebert, and R. Sukthankar, *Beyond local appearance: Category recognition from pairwise interactions of simple features*, CVPR, 2007.

[90] Y. Li, Y.-Z. Song, and S. Gong, *Sketch recognition by ensemble matching of structured features*, in In British Machine Vision Conference (BMVC), 2013.

[91] L. Lin, X. Wang, W. Yang, and J. Lai, *Discriminatively trained and-or graph models for object shape detection*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, PP (2014), pp. 1–1.

[92] D. Lowe, *Distinctive image features from scale-invariant keypoints*, International journal of computer vision, 60 (2004), pp. 91–110.

[93] D. G. Lowe, *Object recognition from local scale-invariant features*, in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, IEEE, 1999, pp. 1150–1157.

[94] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, *Using contours to detect and localize junctions in natural images*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1–8.

[95] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*, (1982).

[96] D. Martin, C. Fowlkes, D. Tal, and J. Malik, *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, in Computer Vision, 2001.Proceedings. Eighth IEEE International Conference on, vol. 2, 2001, pp. 416 –423 vol.2.

[97] J. Matas, O. Chum, M. Urban, and T. Pajdla, *Robust wide-baseline stereo from maximally stable extremal regions*, Image and vision computing, 22 (2004), pp. 761–767.

[98] X. Meng, Z. Wang, and L. Wu, *Building global image features for scene recognition*, Pattern Recognition, 45 (2012), pp. 373 – 380.

[99] T. Minka, *Estimating a dirichlet distribution*, Tech. Report 8, 2003.

[100] F. Mokhtarian and A. Mackworth, *Scale-based description and recognition of planar curves and two-dimensional shapes*, IEEE Trans. Pattern Anal. Mach. Intell., 8 (1986), pp. 34–43.

[101] F. Mokhtarian and A. Mackworth, *A theory of multiscale, curvature-based shape representation for planar curves*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 14 (1992), pp. 789 –805.

[102] D. Munoz, J. A. Bagnell, and M. Hebert, *Stacked hierarchical labeling*, in Proceedings of the 11th European conference on Computer vision: Part VI, ECCV'10, Berlin, Heidelberg, 2010, Springer-Verlag, pp. 57–70.

[103] T. Ojala, M. Pietikainen, and T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24 (2002), pp. 971–987.

[104] A. Oliva and A. Torralba, *Modeling the shape of the scene: A holistic representation of the spatial envelope*, International Journal of Computer Vision, 42 (2001), pp. 145–175.

[105] B. A. Olshausen and D. J. Field, *Sparse coding with an overcomplete basis set: A strategy employed by v1?*, Vision research, 37 (1997), pp. 3311–3325.

[106] A. Perina, N. Jojic, U. Castellani, M. Cristani, and V. Murino, *Object recognition with hierarchical stel models*, in Proceedings of the 11th European conference on Computer vision: Part VI, ECCV'10, Berlin, Heidelberg, 2010, Springer-Verlag, pp. 15–28.

[107] F. Perronnin, J. Sánchez, and T. Mensink, *Improving the fisher kernel for large-scale image classification*, in Computer Vision–ECCV 2010, Springer, 2010, pp. 143–156.

[108] E. Persoon and K.-S. Fu, *Shape discrimination using fourier descriptors*, Systems, Man and Cybernetics, IEEE Transactions on, 7 (1977), pp. 170 –179.

[109] Z. Ping, R. Wu, and Y. Sheng, *Image description with chebyshev-fourier moments*, JOSA A, 19 (2002), pp. 1748–1754.

[110] A. Reeves, R. Prokop, S. Andrews, and F. Kuhl, *Three-dimensional shape analysis using moments and fourier descriptors*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 10 (1988), pp. 937 –943.

[111] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof, *Irregular lattices for complex shape grammar facade parsing*, in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, June 2012, pp. 1640–1647.

[112] H. Rom and G. Medioni, *Hierarchical decomposition and axial shape description*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 15 (1993), pp. 973–981.

[113] E. H. Rosch, *Natural categories*, Cognitive Psychology, 4 (1973), pp. 328 – 350.

[114] E. Rosten and T. Drummond, *Machine learning for high-speed corner detection*, in Computer Vision–ECCV 2006, Springer, 2006, pp. 430–443.

[115] Y. Rui, A. She, and T. Huang, *Modified fourier descriptors for shape representation-a practical approach*, in Proceedings First Int'l Workshop Image Databases and Multi Media Search, vol. 22, 1996, p. 23.

[116] B. C. Russell, J. Sivic, J. Ponce, and H. Dessales, *Automatic alignment of paintings and photographs depicting a 3d scene*, in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 545–552.

[117] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, *Adapting visual category models to new domains*, in European Conference on Computer Vision, 2010, pp. 213–226.

[118] B. Sapp, A. Toshev, and B. Taskar, *Cascaded models for articulated pose estimation*, in ECCV, 2010.

[119] H. Schneiderman and T. Kanade, *A statistical method for 3d object detection applied to faces and cars*, in Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 1, IEEE, 2000, pp. 746–751.

[120] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia, *Symmetry-based indexing of image databases*, J. VISUAL COMMUNICATION AND IMAGE REPRESENTATION, 9 (1998), pp. 366–380.

[121] E. Shechtman and M. Irani, *Matching local self-similarities across images and videos*, in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[122] A. Shokoufandeh, L. Bretzner, D. Macrini, M. Fatih Demirci, C. Jönsson, and S. Dickinson, *The representation and matching of categorical shape*, Computer Vision and Image Understanding, 103 (2006), pp. 139–154.

[123] J. Shotton, A. Blake, and R. Cipolla, *Multiscale categorical object recognition using contour fragments*, TPAMI, (2008).

[124] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros, *Data-driven visual similarity for cross-domain image matching*, in ACM Transactions on Graphics (TOG), vol. 30, ACM, 2011, p. 154.

[125] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, *Shock graphs and shape matching*, International Journal of Computer Vision, 35 (1999), pp. 13–32.

[126] Y. Singer and N. Srebro, *Pegasos: Primal estimated sub-gradient solver for svm*, in ICML, 2007.

[127] P. Smolensky, *Information processing in dynamical systems: Foundations of harmony theory*, (1986).

[128] Y.-Z. Song, P. Arbelaez, P. Hall, C. Li, and A. Balikai, *Finding semantic structures in image hierarchies using laplacian graph energy*, in Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10, Berlin, Heidelberg, 2010, Springer-Verlag, pp. 694–707.

[129] Y.-Z. Song, D. Pickup, C. Li, P. Rosin, and P. Hall, *Abstract art by shape classification*, IEEE Transactions on Visualization and Computer Graphics, 19 (2013), pp. 1252–1263.

[130] Y.-Z. Song, P. L. Rosin, P. M. Hall, and J. Collomosse, *Arty shapes*, in Proceedings of the Fourth Eurographics conference on Computational Aesthetics

in Graphics, Visualization and Imaging, Eurographics Association, 2008, pp. 65–72.

[131] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, *Skeleton based shape matching and retrieval*, in Shape Modeling International, 2003, may 2003, pp. 130 – 139.

[132] M. Szummer and R. Picard, *Indoor-outdoor image classification*, in Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on, 1998, pp. 42–51.

[133] B. Taskar, C. Guestrin, and D. Koller, *Max-margin markov networks*, in NIPS, 2003.

[134] M. Teague, *Image analysis via the general theory of moments*, JOSA, 70 (1980), pp. 920–930.

[135] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, *Shape grammar parsing via reinforcement learning*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, June 2011, pp. 2273–2280.

[136] C.-H. Teh and R. Chin, *On image analysis by the methods of moments*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 10 (1988), pp. 496–513.

[137] A. Temlyakov, B. Munsell, J. Waggoner, and S. Wang, *Two perceptually motivated strategies for shape classification*, in Computer Vision and Pattern Recognition, 2010 IEEE Conference on, june 2010, pp. 2289 –2296.

[138] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool, *Towards multi-view object class detection*, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, IEEE, 2006, pp. 1589–1596.

[139] A. Torralba, K. P. Murphy, and W. T. Freeman, *Sharing features: efficient boosting procedures for multiclass object detection*, in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, IEEE, 2004, pp. II–762.

[140] L. Torresani, V. Kolmogorov, and C. Rother, *Feature correspondence via graph matching: Models and global optimization*, Computer Vision–ECCV 2008, (2008), pp. 596–609.

[141] C.-F. Tsai, *Bag-of-words representation in image annotation: A review*, ISRN Artificial Intelligence, 2012 (2012).

[142] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, *Large margin methods for structured and interdependent output variables*, JMLR, (2005).

[143] Z. Tu, X. Chen, A. Yuille, and S.-C. Zhu, *Image parsing: Unifying segmentation, detection, and recognition*, International Journal of Computer Vision, 63 (2005), pp. 113–140.

[144] S. Ullman, *High-level vision: Object recognition and visual cognition*, MIT press, 2000.

[145] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, *Kernel codebooks for scene categorization*, in Computer Vision–ECCV 2008, Springer, 2008, pp. 696–709.

[146] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, vol. 2, Wiley New York, 1998.

[147] A. Vedaldi and B. Fulkerson, *VLFeat: An open and portable library of computer vision algorithms*, 2008.

[148] A. Vedaldi and A. Zisserman, *Efficient additive kernels via explicit feature maps*, in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010.

[149] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, *Locality-constrained linear coding for image classification*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 3360–3367.

[150] M. Wertheimer, *Laws of organization in perceptual forms*, A source book of Gestalt psychology, (1938), pp. 71–88.

[151] Wikipedia, *Abstract art — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[152] ——, *Chinese painting — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[153] ——, *Impressionism — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[154] ——, *Japanese painting — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[155] ——, *Korea painting — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[156] ——, *Modernism — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[157] ——, *Outsider art — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[158] ——, *Painting — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[159] ——, *Photorealism — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[160] ——, *Surrealism — Wikipedia, the free encyclopedia*, 2015. [Online; accessed 14-Mar-2015].

[161] J. Winn and J. Shotton, *The layout consistent random field for recognizing and segmenting partially occluded objects*, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, IEEE, 2006, pp. 37–44.

[162] Q. Wu, H. Cai, and P. Hall, *Learning graphs to model visual objects across different depictive styles*, in Computer Vision–ECCV 2014, Springer, 2014, pp. 313–328.

[163] Q. Wu and P. Hall, *Prime shapes in natural images*, in Proceedings of the British Machine Vision Conference, BMVA Press, 2012, pp. 45.1–45.12.

[164] Q. Wu and P. Hall, *Modelling visual objects invariant to depictive style*, in In Proceeding of the British Machine Vision Conference 2013 (BMVC 2013), BMVA Press, 2013.

[165] S. Xia and E. R. Hancock, *Learning class specific graph prototypes*, in Image Analysis and Processing–ICIAP 2009, Springer, 2009, pp. 269–277.

[166] B. Xiao, Y.-Z. Song, A. Balika, and P. M. Hall, *Structure is a visual class invariant*, in Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2008, pp. 329–338.

[167] B. Xiao, S. Yi-Zhe, and P. Hall, *Learning invariant structure for object identification by using graph methods*, Computer Vision and Image Understanding, 115 (2011), pp. 1023–1031.

[168] Y. Yang and D. Ramanan, *Articulated pose estimation with flexible mixtures-of-parts*, in IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1385–1392.

[169] B. Yao and L. Fei-Fei, *Action recognition with exemplar based 2.5d graph matching*, in ECCV, 2012.

[170] Z. Ying and D. Castañon, *Partially occluded object recognition using statistical models*, International Journal of Computer Vision, 49 (2002), pp. 57–78.

[171] D. Zhang and G. Lu, *Evaluation of mpeg-7 shape descriptors against other shape descriptors*, Multimedia Systems, 9 (2003), pp. 15–30.

[172] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, *Image classification using super-vector coding of local image descriptors*, in Computer Vision–ECCV 2010, Springer, 2010, pp. 141–154.

[173] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille, *Part and appearance sharing: Recursive compositional models for multi-view*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1919–1926.

[174] L. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille, *Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion*, in Computer Vision–ECCV 2008, Springer, 2008, pp. 759–773.